

中图分类号:

论文编号: 10357WA21301037

安徽大学

硕士学位论文

基于可靠性分析的主板  
功能测试策略研究

作者姓名	李瑶瑶
专业学位类别	电子信息
专业学位领域	人工智能
指导教师	陈思宝、赵云波

**Research on the Board-level  
Functional Test Strategy Based on Reliability Analysis**

A Dissertation Submitted for the Degree of Master

**Candidate: LI Yaoyao**

**Supervisor: CHEN Sibao、ZHAO Yunbo**

中图分类号:

论文编号: 10357WA21301037

## 硕士学位论文

# 基于可靠性分析的主板 功能测试策略研究

作者姓名	李瑶瑶	申请学位级别	工程硕士
指导教师姓名	陈思宝、赵云波	职 称	教授、教授
专业名称	人工智能	研究方向	笔记本主板功能测试
学习时间自	2021 年 9 月 7 日	起至	2024 年 6 月 30 日止
论文提交日期	2024 年 3 月 26 日	论文答辩日期	2024 年 5 月 17 日

# 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得安徽大学或其他教育机构的学位或证书而使用过的材料。如有学术不端行为，一切后果由本人承担，与导师和安徽大学无关。

学位论文作者签名：李瑶瑶

签字日期：2024年5月17日

# 学位论文版权使用授权书

本学位论文作者完全了解安徽大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权安徽大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：李瑶瑶

导师签名：李冰 陈思宝

签字日期：2024年5月17日

签字日期：2024年5月17日

## 摘 要

电子产品的功能大多集成在主板上，因此为保障电子产品的出厂质量，需要对主板质量进行严格把控。主板功能测试是保证主板质量的关键环节，然而为使得测试的功能更加全面，往往需要花费较高的测试成本。为减少测试成本，本文对主板进行了可靠性分析，并且在此基础上提出了一种基于可靠性分析的主板功能测试策略设计方法，该方法能够同时兼顾测试质量和测试成本，从而提升笔记本测试环节的整体经济效益。本文的主要内容包括：

第一，针对主板的维修数据缺乏导致难以对主板进行准确地可靠性分析的问题，本文提出了一种基于故障树的主板可靠性分析方法，该方法充分的考虑了元件间的互连信息，能够准确地分析主板的可靠性。首先以某典型笔记本主板为例，为该主板建立了面向主板功能测试环节的故障树，然后针对主板维修数据充足和缺乏的两个场景分别设计了自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法。特别的，为了提升可靠性分析的准确性，本文在建立故障树模型时，充分地考虑了元件间互连信息。实验表明，包含元件互连信息的故障树对主板的可靠性分析更加准确，并且所提方法能够用于测试策略设计，保证了主板测试质量，同时降低了主板测试时间成本。

第二，针对行业现行的主板功能测试策略设计方法可解释性差，并且往往存在过拟合的问题，本文提出了一种融合决策树和可靠性分析的可解释测试策略设计方法，该方法能够提升主板功能测试环节的整体经济效益，并且具有较强的可解释性，能够为主板生产过程提出针对性的意见，进一步提升主板的良品率。首先利用主板可靠性分析方法获得主板中各测项的可靠性指标范围。然后在该指标范围内生成第一层决策树模型的训练数据，根据训练出的决策树模型缩小各测项的可靠性指标范围并进行特征选择。接下来，根据第一层决策树模型的训练结果准备第二层决策树的训练数据，并根据训练结果为主板设计测试策略。实验结果表明，所提方法具有较强可解释性，并且能够充分利用测试结果中的信息，显著提升了整体经济效益。

**关键词：**主板功能测试，测试策略设计，可靠性分析，决策树模型

## Abstract

The majority of electronic products' functions are integrated on the motherboard, so in order to ensure the quality of electronic products, the quality of the motherboard needs to be strictly controlled. The board-level functional test aims to ensure a more comprehensive test function, but often requires a higher testing cost. In order to reduce the test cost, this thesis performs a reliability analysis on the motherboard and proposes a board-level function test strategy based on reliability analysis. This approach optimizes both the quality of the test and cost-efficiency, enhancing the overall economic benefits of the laptop testing process. The main content of the thesis are as follows:

First, to address the problem that the lack of repair data on the motherboard makes it difficult to accurately assess the reliability of the motherboard, a fault tree analysis-based methodology for evaluating circuit board reliability is proposed, which fully takes into account the interconnection information between the components and can accurately assess the reliability of the motherboard. Firstly, taking a typical laptop motherboard as an example, a fault tree oriented to the board-level functional test is established. Following this, bottom-up and top-down motherboard reliability analysis methods are developed for scenarios with both sufficient and insufficient maintenance data for the motherboard. Importantly, our method incorporates detailed component interconnection information during the establishment of the fault tree model, thereby improving the accuracy of reliability assessments. Experiments show that the fault tree with component interconnection information is more accurate for the circuit reliability analysis, and the proposed method can ensure the quality of circuit board functional testing and reduce the time cost of the process.

Second, for the industry's current board-level functional test strategy design methods are poorly interpretable and often has the overfitting problem, this thesis proposes an interpretable test strategy design method that integrates the decision tree and reliability analysis, which can improve the overall economic efficiency of the board-level functional test process, and has a strong interpretability, which can provide targeted advice for the motherboard production

process and further improve the motherboard yield rate. Firstly, the reliability analysis method of the circuit board is used to obtain the reliability index range of each test item in the motherboard. Then, the training data of the first layer decision tree model is generated within the index range, and the reliability index range of each test item is narrowed and feature selection is performed according to the trained decision tree model. Next, the training data of the second layer decision tree is prepared based on the training results of the first layer decision tree model, and the test strategy is designed for the motherboard based on the training results. The experimental results show that the proposed method has strong interpretability and can make full use of the information in the test results, which significantly improves the overall economic efficiency.

**Key words:** Board-level functional test, Test strategy design, Reliability analysis, Decision tree model

# 目 录

<b>第一章 绪论</b> .....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	1
1.2.1 主板可靠性分析.....	2
1.2.2 主板功能测试策略设计.....	2
1.3 本文的工作与章节安排.....	5
1.3.1 主要工作.....	5
1.3.2 章节安排.....	5
<b>第二章 相关基础知识</b> .....	6
2.1 故障树分析方法.....	7
2.2 决策树建模方法.....	8
2.3 本章小结.....	10
<b>第三章 基于元件互连信息的主板可靠性分析方法</b> .....	11
3.1 主板故障树的建立.....	11
3.1.1 普通的主板故障树建立.....	11
3.1.2 包含元件互连信息的主板故障树建立.....	16
3.2 主板可靠性分析方法.....	19
3.2.1 自底向上的主板可靠性分析方法.....	19
3.2.2 自顶向下的主板可靠性分析方法.....	20
3.3 基于可靠性分析的主板功能测试策略设计.....	22
3.3.1 整体流程.....	22
3.3.2 测项的可靠性指标设定.....	22
3.3.3 主板功能测试策略设计.....	23
3.4 数据集及评价指标.....	23
3.4.1 笔记本主板数据集.....	23
3.4.2 测试策略的评价指标.....	25

3.5 实验与分析 .....	26
3.5.1 数据准备 .....	26
3.5.2 实验过程与分析 .....	27
3.6 本章小结 .....	32
<b>第四章 融合决策树和可靠性分析的可解释测试策略设计方法 .....</b>	<b>34</b>
4.1 方法介绍 .....	34
4.1.1 整体流程 .....	34
4.1.2 基于决策树的主板功能测试策略设计方法 .....	36
4.1.3 融合决策树和可靠性分析的主板功能测试策略设计方法 .....	40
4.1.4 融合双层决策树和可靠性分析的主板功能测试策略设计方法 .....	40
4.2 实验与分析 .....	42
4.2.1 数据准备 .....	42
4.2.2 参数设置 .....	45
4.2.3 实验过程和结果分析 .....	46
4.3 本章小结 .....	47
总结与展望 .....	48
参考文献 .....	50
攻读硕士学位期间取得的学术成果 .....	56
攻读硕士学位期间参与的科研项目 .....	57
致 谢 .....	58

# 图目录

图 1.1	本文结构图.....	6
图 2.1	故障树示意图.....	7
图 2.2	决策树模型的示意图.....	9
图 3.1	某典型笔记本主板的电路图.....	11
图 3.2	键盘功能模块的故障树分支.....	14
图 3.3	某典型笔记本主板的故障树.....	15
图 3.4	以键盘功能测项故障为顶事件的故障树分支.....	17
图 3.5	笔记本主板的故障树(包含元件互连信息).....	18
图 3.6	底事件分类后的笔记本主板的故障树.....	20
图 3.7	测试策略设计过程.....	22
图 4.1	融合决策树和可靠性分析的可解释测试策略设计方法流程图.....	35
图 4.2	主板测项所组成的集合 $M$ 、 $M_1$ 和 $M_2$ 的关系.....	35
图 4.3	基于决策树的阈值设定方法的流程图.....	36

## List of Figures

Fig. 1.1	Structure of this thesis.....	6
Fig. 2.1	The example diagram of fault tree .....	7
Fig. 2.2	An example diagram of a decision tree model.....	9
Fig. 3.1	The circuit diagram of a typical laptop motherboard.....	11
Fig. 3.2	The fault tree branching for the keyboard function module .....	14
Fig. 3.3	The fault tree of a typical laptop motherboard.....	15
Fig. 3.4	Fault tree branching with keyboard functional test item failure as top event .....	17
Fig. 3.5	Fault tree of a laptop motherboard (with component interconnection information)	18
Fig. 3.6	The fault tree of a laptop motherboard after bottom event classification .....	20
Fig. 3.7	Flow of functional test strategy design method based on reliability analysis.....	22
Fig. 4.1	The flowchart of the interpretable test strategy design method based on decision tree and reliability analysis.....	35
Fig. 4.2	The relationship between the sets M, M1 and M2 formed by the test items .....	35
Fig. 4.3	The flowchart of the decision tree based threshold setting approach .....	36

# 表目录

表 3.1	某典型笔记本主板的功能测试项目 .....	12
表 3.2	笔记本主板故障树的中间事件描述 .....	13
表 3.3	笔记本主板故障树的底事件描述 .....	14
表 3.4	某典型笔记本主板中各测项的故障原因集合 .....	16
表 3.5	包含元件互连信息的笔记本主板故障树的底事件描述 .....	17
表 3.6	某典型笔记本主板中各测项的故障原因集合(包含元件互连信息) .....	19
表 3.7	基本事件分类结果 .....	20
表 3.8	基本事件发生概率成比例时各测项的不良率表达式 .....	21
表 3.9	某典型笔记本主板中各测项的真实良率和平均测试时间 .....	24
表 3.10	实际数据Dinit和上Dsteady各测项的不良率和平均测试时间 .....	26
表 3.11	仿真数据D'init1和D'steady1上各测项的良率 .....	27
表 3.12	3 种测试策略设计方法对应的测试策略 .....	28
表 3.13	3 种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平 ...	28
表 3.14	基本事件发生概率 .....	29
表 3.15	自底向上和自顶向下的主板可靠性分析方法对应的不良率阈值 .....	30
表 3.16	4 种测试策略设计方法所对应的测试策略 .....	31
表 3.17	4 种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平 ...	32
表 4.1	底事件的发生概率范围 .....	43
表 4.2	分别通过统计方法和机理方法获得的初始良率阈值设定范围 .....	44
表 4.3	4.1.1 节、4.1.2 节和 4.1.3 节所提方法的样本聚类结果 .....	45
表 4.4	4 种测试策略设计方法对应的测试策略 .....	46
表 4.5	4 种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平 .....	46

## List of Tables

Tab. 3.1	Functional test items for a typical laptop motherboard.....	12
Tab. 3.2	The description of intermediate events in the fault tree of a laptop motherboard...	13
Tab. 3.3	Bottom event description of the laptop motherboard fault tree.....	14
Tab. 3.4	Collections of failure causes for each test item in a typical laptop motherboard....	16
Tab. 3.5	Bottom event description of the laptop motherboard fault tree containing component interconnect information.....	17
Tab. 3.6	Failure cause sets for each test in a typical laptop motherboard (with component interconnect information).....	19
Tab. 3.7	The results of basic event classification .....	20
Tab. 3.8	The expression for the failure rate of each test item when the occurrence probability of the basic event is proportional.....	21
Tab. 3.9	The true yield and average test time for each test item in a typical laptop motherboard .....	24
Tab. 3.10	The failure rate and the average test time of each test item on $D_{init}$ and $D_{steady}$ .....	26
Tab. 3.11	The yields of each test item on simulation data $D'_{init1}$ and $D'_{steady1}$ .....	27
Tab. 3.12	Test strategies corresponding to the 3 test strategy design methods .....	28
Tab. 3.13	The average total test cost, average test time and defect level corresponding to the 3 test strategy design methods.....	28
Tab. 3.14	The occurrence probability of basic events.....	29
Tab. 3.15	Failure rate thresholds corresponding to the bottom-up and the top-down motherboard reliability analysis methods .....	30
Tab. 3.16	Test strategies corresponding to the 4 test strategy design methods .....	31
Tab. 3.17	The average total test cost, the average test time and the defect level corresponding to the four test strategy design approaches.....	32
Tab. 4.1	Probability ranges for the occurrence of bottom events.....	43
Tab. 4.2	Initial yield threshold setting range obtained by statistical and mechanistic methods	

.....	44
Tab. 4.3 Sample clustering results for the methods proposed in subsections 4.1.1, 4.1.2 and 4.1.3.....	45
Tab. 4.4 Test strategies corresponding to the 4 test strategy design methods .....	46
Tab. 4.5 The average total test cost, average test time and defect level corresponding to the 4 test strategy design methods.....	46

# 第一章 绪论

## 1.1 研究背景与意义

近年来，随着科技的发展，电子产品为人类的生活带来了巨大便利。与此同时，人们生活水平逐渐提升，消费者对电子产品质量的要求越来越高，电子制造行业迎来更大挑战。为保证电子产品的出厂质量，在出厂前会对电子产品进行整机测试。随着集成电路的发展，电子产品的核心功能大多集成在主板上，所以主板质量直接影响电子产品的质量<sup>[1]</sup>。

通过主板测试能够检测出主板的故障，为快速定位和维修主板故障，主板测试过程分为多个阶段：自动光学检测、自动 X 射线检测、结构测试和主板功能测试<sup>[2-4]</sup>。其中，自动光学检测和自动 X 射线检测能够可靠地检验焊点缺陷；结构测试用来验证单个元件的性能；主板功能测试在捕捉结构测试不易发现的缺陷方面发挥着重要作用，能够检测整个系统的可靠性，所以主板功能测试是保障主板质量的关键环节。

随着计算机主板的复杂度越来越高，计算机主板的测试过程越来越复杂。在测试过程中为尽可能覆盖主板的功能，从而保证主板质量，需要花费较高的测试成本<sup>[5-7]</sup>，因此在实际测试过程中需要同时考虑主板的测试质量和测试成本。

当前行业侧重于保证主板的测试质量，以防不良主板通过功能测试环节的测试从而影响笔记本的品质。工厂在主板功能测试环节加入大量的测试项目，以此增加对主板功能进行测试的覆盖率，从而拦截不良主板，减少不良主板流入笔记本整机测试环节。然而测试项目的增加必然会导致测试时间成本和人力成本的增加，从而影响测试产能以及整体经济效益，这一矛盾是生产测试工程以及测试开发人员亟待解决的难题。因此为主板功能测试环节设计兼顾测试质量和测试成本的测试策略至关重要<sup>[8]</sup>。

## 1.2 国内外研究现状

本文主要研究的是基于可靠性分析的主板功能测试策略，本节主要回顾了近些年来为主板功能测试阶段设计测试策略的国内外研究现状，将从主板可靠性分析和主板功能测试策略设计两个任务来展开介绍。

### 1.2.1 主板可靠性分析

对系统进行可靠性分析的方法主要有：可靠性框图法、故障树分析、Markov 分析、Monte-Carlo 仿真、GO 法和 Petri 网<sup>[9]</sup>。其中，故障树分析方法能够同时进行定性分析和定量计算，广泛应用于主板可靠性分析领域。

目前基于故障树模型的主板可靠性分析的相关研究大多集中于对生产环节中可变因素的分析，如材料品质、人为因素和设备状态等。比较有代表性的工作有：Shu 等学者<sup>[10]</sup>提出了一种基于直觉模糊数的故障树分析算法，该算法集成了专家的知识和技术，提供了底层事件可能发生故障的可能性；Wang 等学者<sup>[11]</sup>基于 Shu 等学者建立的故障树研究了三角直觉模糊数的新算术运算和逻辑运算，并将它们应用于印刷电路板装配系统的故障分析。

上述研究中的主板可靠性分析过程往往忽略了主板结构这一不可变因素。主板结构中主要包括元件互连<sup>[12]</sup>和元件组成<sup>[13-15]</sup>两大信息，有少量研究分析系统可靠性时初步考虑了结构信息，但均不全面：Yokogawa 等学者<sup>[12]</sup>通过建立故障树对半导体片上互连系统的可靠性进行分析，建树过程中细致分析了元件间的连接故障，但是片上互连系统故障仅是引发印刷电路板（printed circuit board, PCB）故障的部分原因，因而有必要在该项工作的基础上探索影响 PCB 可靠性的更多因素；文献<sup>[13]</sup>基于电路原理图为自动变速器控制系统建立故障树，认为 CPU 中各元件自身故障是造成系统故障的全部原因，但是忽略了元件间的连接故障，并且建树过程仅分析了系统的核心组件 CPU 模块，忽略了系统的其他组成部分故障的可能性；Hedel 等学者<sup>[16]</sup>提出了一种使用故障树分析来评估自动驾驶电动汽车可靠性的先进方法，通过分析转向系统、自动化系统和动力系统等关键子系统对自动驾驶电动汽车的影响，以保持其安全运行；文献<sup>[14,15]</sup>利用电动车的动力系统结构示意图为其建立故障树，该研究认为系统的可靠性仅由元件或子元件特性决定，忽略了元件间的连接信息，并且并未深入分析系统的主要元件 PCB 的可靠性。上述研究大都忽略了元件间的互连故障，影响了可靠性分析的准确性。因此，有必要在主板可靠性分析过程中考虑元件互连信息。

### 1.2.2 主板功能测试策略设计

目前主要通过测项排序<sup>[17-19]</sup>和测项选择<sup>[20-24]</sup>两种方法来设计测试策略。其中，测项排序方法是为测项设定一个测试顺序，将容易发生故障的测项的测试顺序提前，检测

到不良测项时即可停止对该块主板的检测，认定该块主板为不合格，所以测项排序方法能够降低不良主板的测试成本，对于合格主板，需要对所有测项进行测试。在测项排序方法中比较有代表性的工作是法国国家计算机与自动装置研究院的 Akkouche 团队<sup>[17-19]</sup>研究了集成电路生产测试初期阶段的测项排序问题，提出了一种只使用一小部分功能电路数据的测项排序方法。

测项选择方法通过测项相关性和最优化建模等方法减少冗余测项，仅对主板中部分测项进行测试，所以能够同时降低不良主板和合格主板的测试成本<sup>[7,25-28]</sup>。由于工厂中主板的生产过程具有大批量和高良率的特征，测项选择方法更适用于主板功能测试环节的测试策略设计。

目前关于主板功能测试的测项选择方法均是基于数据驱动模型设计测试策略<sup>[29-32]</sup>。例如，Feige 等学者<sup>[29,33]</sup>直接使用贪心算法，能够最大程度找出缺陷样本，但是由于主板生产具有高良率、大批量等特征，导致该算法在主板功能测项选择上存在过拟合问题，并且需要消耗大量的时间和内存成本；Pan 等学者虽然<sup>[30,34]</sup>通过贝叶斯网络获得测项相关性信息，并在不相关测项上使用最优化建模方法，有效的避免了过拟合问题，但是优化模型的目标是最大程度找出缺陷样本，这仅考虑了测试质量，忽略了测试成本，并未考虑测试环节的整体经济效益；与此同时，Liu 等学者<sup>[31]</sup>在预测芯片质量的基础上，根据测项的测试结果分布为不同质量水平的芯片设计了对应的测项选择方法，通过这种方法，可以在保证测试效果的同时，显著降低测试成本，但是该项研究主要降低了良率较低芯片的测试成本，对整体经济效益的降低十分有限。紧接着，Liu 等学者<sup>[35]</sup>提出一种基于机器学习的自适应测项选择方法，可以动态调整测试过程中的测试项目，从而降低测试成本。由于这些方法未关注主板生产、测试等过程中的机理信息，未对主板进行可靠性分析，忽略了造成主板不良的根本原因，设计出的测试策略往往存在过拟合和可解释性差的问题。

主板的可靠性分析过程包含了丰富的机理信息，然而目前主板的可靠性分析结果主要是为主板生产环节提供参考信息，例如，找出影响主板可靠性的重要原因，并对此进行改进，从而提升主板的可靠性<sup>[36,37]</sup>，尚未有研究将主板可靠性分析结果用于主板功能测试环节的测试策略设计。然而通过拦截指标参数超出可靠性范围的测项，保障主板流向下一环节的合格率，是一种科学的测试策略设计方法。因此，对主板进行可靠性分析，

将机理信息用于测试策略设计是有效的应对方法。

基于可靠性分析的测试策略设计方法是从机理出发分析主板的可靠性，能够在主板测试数据缺乏的情况下保证主板功能测试阶段的测试质量，但在数据充足的情况下，难以充分利用历史数据，提升整体经济效益；同时纯数据驱动建模方法的建模结果易受数据波动影响，容易过拟合，但是能够利用最新数据获得针对目标的最优结果。因此，在主板测试数据充足的情况下，有必要在机理分析的基础上融合数据驱动建模，进一步提升整体经济效益。

目前，已有大量研究将机理分析方法与数据驱动建模相融合，用于医疗药物的毒性评估<sup>[38,39]</sup>、废弃电池的剩余能量评估<sup>[40]</sup>等领域。由于我们研究的问题是如何寻找整体经济效益最优所对应的测试策略，而决策树模型可以确定一个最可能达到目标的策略，因而对我们的研究具有重要参考价值。

决策树模型作为一种数据驱动模型，在分类<sup>[41-43]</sup>和预测<sup>[44,45]</sup>方面都发挥了重要的作用，被广泛应用于电力生产过程中可控变量的调节<sup>[46]</sup>、主板故障诊断<sup>[47,48]</sup>和芯片质量预测<sup>[31]</sup>等领域。张俊玉等学者提出利用 CART 决策树方法设置影响煤耗的关键可控变量和相关参数的调节范围<sup>[46]</sup>，相比于传统的关联规则算法，该方法在计算效率和降低煤耗方面都具有一定优越性。Ye 等学者<sup>[47]</sup>提出了一种基于决策树的自适应的主板诊断方法，该方法保证了诊断的准确性，同时，显著降低了诊断时所需的不良症状数量，有效地降低了诊断成本。Ye 等学者<sup>[48]</sup>在前一项工作的基础上进一步研究，提出了一种基于增量决策树的自适应诊断方法，该诊断系统可以及时适应新错误场景的发生。

但是上述工作大多是将决策树模型用于主板故障的定位和诊断，有部分学者将决策树模型用于软件<sup>[49-51]</sup>或模拟电路<sup>[23,52,53]</sup>的测试策略设计，却均未从提升整体经济效益的角度设计模型。将整体经济效益作为决策树模型的决策目标时，决策树图像能够展现决策与结果间的映射关系。具体的，决策树路径能够直接对应测试策略，并且能够直观的反映出测试策略与经济效益的对应关系；二叉决策树的结构和目标测试策略的规则能够很好的对应起来。所以决策树模型能够选择一条整体经济效益最优的测试策略，同时具有较强的可解释性，适合作为提升主板整体经济效益的数据驱动模型。此外，与纯机理方法相比，将决策树模型和主板可靠性分析方法相融合能够更加充分的利用现场数据中的有效信息，针对性的设计科学高效的测试策略，对于提升整体经济效益具有重要意义。

## 1.3 本文的工作与章节安排

### 1.3.1 主要工作

本文以笔记本电脑功能测试产线的主板功能测试环节为研究对象，重点提升测试环节的整体经济效益。首先系统分析主板的电路结构，对主板进行可靠性分析。然后为主板功能测试环节设计兼顾测试质量和测试效率的测试策略。

工作内容一：基于元件互连信息的主板可靠性分析。首先针对现有的主板可靠性分析方法往往未充分考虑元件互连信息的问题，通过系统的分析笔记本主板电路结构，确定引发主板功能测试故障的原因，建立包含电子元件互连信息的故障树。然后针对主板维修数据充足和缺乏的两个场景分别设计了自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法，利用故障树对主板的可靠性进行定性分析和定量计算。最后，将主板可靠性分析结果用于测试策略设计，用以验证主板可靠性分析的准确性以及基于可靠性分析设计测试策略的可行性。

工作内容二：融合决策树和可靠性分析的可解释测试策略设计方法研究。现有的测试策略设计方法缺乏可解释性，并且由于未充分利用机理信息，使得测试策略的设计往往存在过拟合的问题，为解决这一问题，本文提出了一种可解释的测试策略设计方法，该方法将可靠性分析和决策树模型相互融合设计出最佳的测试策略。首先利用主板可靠性分析方法得到主板中各测项的可靠性指标范围，从而缓解测试策略的过拟合问题。然后根据不同的测试要求，选择相应的样本聚类方式，准备决策树模型的训练数据，并训练模型。最后，通过训练出的模型得到整体经济效益最优或者测试效率最高所对应的关键参数取值，并设定合适的测试策略。

### 1.3.2 章节安排

本文的文章结构如图 1.1 所示，论文各个章节的安排如下：

第一章，绪论。首先以工厂中某典型笔记本主板为例，对主板功能测试环节进行介绍，阐述为主板功能测试环节设计测试策略的意义。然后从主板可靠性分析和主板功能测试策略设计两方面综述目前的国内外研究现状。最后，提出本文的研究内容。

第二章，相关基础知识。详细介绍了可靠性理论中的故障树分析方法和机器学习领域中的决策树模型的相关知识。

第三章，基于元件互连信息的主板可靠性分析方法。针对现有的电路板可靠性分析方法往往未充分考虑元件间互连信息并且难以用于主板功能测试环节中测试策略的设计的问题，提出了包含原件互连信息的主板可靠性分析方法。首先以某典型笔记本主板为例，为该主板建立了包含元件互连信息的故障树。然后，针对主板维修数据充足和缺乏的两个场景分别设计了自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法。接着，分别基于普通故障树和基于包含元件互连信息故障树，利用所提出的主板可靠性分析方法对某典型笔记本主板的可靠性进行分析，将主板中各测项的可靠性结果用于主板功能测试阶段中测试策略的设计，验证了包含元件互连信息的故障树对主板的可靠性分析更加准确。最后，将上述测试策略设计过程和行业内现行的测试策略设计方法比较，验证了将主板可靠性分析结果用于主板功能测试策略设计的可行性。

第四章，融合决策树和可靠性分析的可解释测试策略设计方法。针对现有的主板功能测试策略设计方法的过拟合和可解释性差的问题，本章提出了一种融合决策树和可靠性分析的可解释测试策略设计方法。首先利用第三章中所提出的主板可靠性分析方法得到某典型笔记本主板中各测项的可靠性指标范围。然后在该指标范围内生成第一层决策树模型的训练数据，并根据训练出的决策树模型缩小各测项的可靠性指标范围并对进行特征选择。接下来，根据第一层决策树模型的训练结果准备第二层决策树的训练数据，并根据训练结果设计主板的测试策略。最后将所提方法和行业现行的测试策略设计方法比较，并完成消融实验，验证了所提方法的有效性。

最后，总结与展望。对本文的主要内容和创新点进行总结，并对未来工作进行展望。

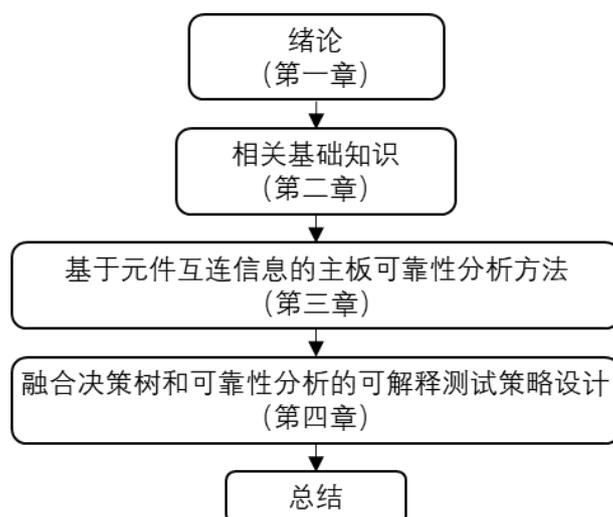


图 1.1 本文结构图

Fig. 1.1 Structure of this thesis

## 第二章 相关基础知识

### 2.1 故障树分析方法

故障树分析方法是一种常见的可靠性分析方法，本文使用该方法分析笔记本主板的可靠性<sup>[54-57]</sup>。故障树一般由三种事件和两个逻辑符号构成，反应了事件间的逻辑因果关系。其中，三种事件分别是顶事件、中间事件和基本事件，两个逻辑符号分别是与门和或门。故障树的建立是从上往下的分析过程，顶事件是系统不希望出现的状态，再分析导致顶事件发生的原因，这些原因构成中间事件，进一步分析导致中间事件发生的原因，直至找到所有中间事件发生的根本原因，这些根本原因即为基本事件。下层事件通过逻辑符号和上层事件相连，若下层事件同时发生会导致上层事件的发生，则通过与门连接；若下层任一事件发生就会导致上层事件的发生，则通过或门连接。

若故障树中下层事件均独立，且下层事件的发生概率已知，可以定量计算上层事件的发生概率。两个基本运算公式如下：

(1) 与门结构：

$$P_X = \prod_{i=1}^n P_{X_i} = \prod_{i=1}^n P_{X_i}, \quad (2.1)$$

其中，上层事件 $X$ 通过与门与下层事件 $X_i$ 相连。

(2) 或门结构：

$$P_X = \bigcup_{i=1}^n P_{X_i} = 1 - \prod_{i=1}^n (1 - P_{X_i}), \quad (2.2)$$

其中，上层事件 $X$ 通过或门与下层事件 $X_i$ 相连。

假设某系统不希望发生的事件为 $T$ ，且系统中基本事件间相互独立。当基本事件 $X_1$ 和中间事件 $M_1$ 同时发生时，顶事件 $T$ 发生；基本事件 $X_1$ 或 $X_2$ 发生时，均会导致中间事件 $M_1$ 发生。则该系统可以建立如图 2.1 所示的故障树。

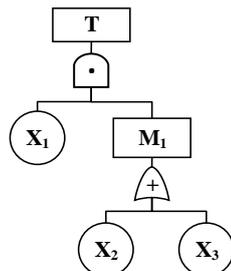


图 2.1 故障树示意图

Fig. 2.1 The example diagram of fault tree

每棵故障树对应一个布尔表达式，布尔表达式中用“ $\cap$ ”表示故障树中的与门，用“ $\cup$ ”表示故障树中的或门。利用化简公式将布尔表达式化简，则图 2.1 中故障树可以表示为如公式(2.3)所示的布尔表达式。

$$\begin{aligned} T &= X_1 \cap M_1 \\ &= X_1 \cap (X_1 \cup X_2) \\ &= (X_1 \cap X_1) \cup (X_1 \cap X_2) \\ &= X_1 \cup (X_1 \cap X_2) \end{aligned} \quad (2.3)$$

用 $P_{X_i}$ 表示底事件 $X_i$ 的发生概率，则顶事件 $T$ 的发生概率可表示为公式(2.4)。

$$\begin{aligned} P_T &= 1 - (1 - P_{X_1})(1 - P_{X_1 \cap X_2}) \\ &= 1 - (1 - P_{X_1})(1 - P_{X_1} \cdot P_{X_2}) \end{aligned} \quad (2.4)$$

## 2.2 决策树建模方法

决策树模型在诸多领域中广泛应用，是一个强大的学习和分类的工具<sup>[44,58-61]</sup>。一个决策树包含两种类型结点：中间节点和叶节点。中间节点包含特征的信息，不同的特征表现，会走向不同的分支，叶节点不再分支且包含类别信息。

决策树可以解决连续性和离散问题，当特征是连续性特征时，中间节点包含的信息如下所示：

$$x_{cont} \leq c, \quad (2.5)$$

其中， $x_{cont}$ 代表连续特征， $c$ 是特征 $x_{cont}$ 连续取值范围中的一个分界值。

当特征是离散特征时，中间节点包含的信息如下所示：

$$x_{disc} = d, \quad (2.6)$$

其中， $x_{disc}$ 代表连续特征， $d$ 是特征 $x_{disc}$ 离散取值中的一个。

假设一批主板有 $r$ 个测试项目，则这些测试项目所对应的良率阈值范围可以表示为 $T = \{T_1, T_2, \dots, T_r\}$ 。若该批主板在不同的阈值组合下进行功能测试所花费的测试成本可以聚成 $p$ 类，用集合 $V = \{V_1, V_2, \dots, V_p\}$ 表示，则决策树的叶节点都对应一个聚类类别。决策树路径上，各中间节点包含的内容是某测项的阈值设定范围，用 $\tilde{T} = \{\tilde{T}_1, \tilde{T}_2, \dots, \tilde{T}_q\}$ 表示，其中， $q$ 为决策树里中间结点的数目。

若各测项的阈值根据中间结点提供的信息设定，即可达到路径末端叶节点所标示的测试成本类别。能够用于测项的阈值设定的决策树模型结构如图 2.2 所示。

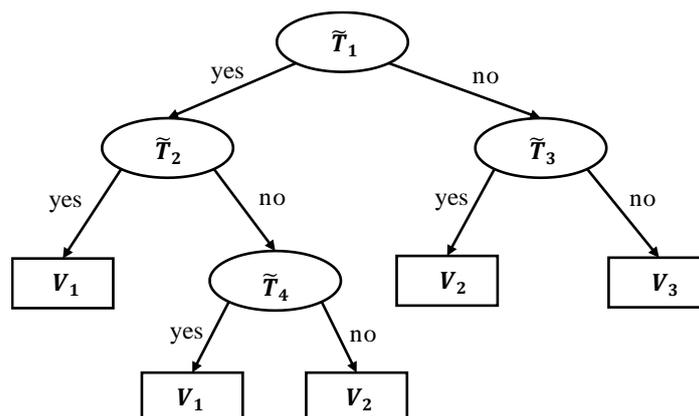


图 2.2 决策树模型的示意图

Fig. 2.2 An example diagram of a decision tree model

图 2.2 中，有两组阈值组合会导致预测出测试成本为 $V_1$ ：按照 $\tilde{T}_1$ 和 $\tilde{T}_2$ 设定测项的阈值可以得到测试成本为 $V_3$ 的情况；此外，按照 $\tilde{T}_1$ 、 $\tilde{T}_2$ 和 $\tilde{T}_4$ 设定测项的阈值同样可以得到测试成本为 $V_3$ 的聚类类别。我们可以将两种良率阈值设定组合进行比较，挑选出最适合的阈值设定组合。

决策树的分类只会考虑少数属性，如图 2.2 所示，我们只需要根据 $\tilde{T}_1$ 和 $\tilde{T}_2$ 提供的信息对两个测项的良率阈值进行设定们就可以得到测试成本为 $V_3$ 的情况。路径上并不会出现所有测项的阈值范围，这意味这些测项测与不测对该测试成本的影响远比在路径上出现的测项要小，因此我们不仅能通过路径上中间结点信息设定测项的良率阈值，还能筛选出不重要的测项。对于这些测项，我们可以直接都不进行测试，或者都进行测试，找到这其中有一个策略的优点，就可以更加充分地利用决策树的特性。

决策树是一个自顶向下增长的贪婪算法，在每个结点上选取分类效果最好的属性，这个过程会持续到这棵树能完美分类所有的训练样例或使用了所有的属性。若不进行剪枝，会出现过拟合现象，目前一般通过先剪枝和后剪枝两种方法缓解决策树的过拟合现象。我们可以通过限制决策树叶节点样本个数进行先剪枝，也可以使用代价复杂度等方法进行后剪枝<sup>[62-65]</sup>。

相比于人工方法，决策树方法避免了复杂的知识获取过程，它可以自动地从历史数据中挖掘出有用的信息，不需要对主板复杂的功能结构非常熟悉，不依赖于专家经验，有很强的分析和学习能力；此外，相比于神经网络，决策树具有很强的可解释性，中间结点和叶节点间有可解释的关系<sup>[66,67]</sup>。

## 2.3 本章小结

本章介绍了可靠性分析方法和决策树建模方法的相关基础知识。其中，可靠性分析方法中重点介绍了故障树分析方法的定量计算过程，在决策树建模方法中介绍了决策树的基本构成和决策树示意图，分析了决策树模型中路径以及结点的含义，并将决策树与人工方法、神经网络算法相对比，突出了决策树模型的优势。

### 第三章 基于元件互连信息的主板可靠性分析方法

在主板测试初期，主板的维修数据缺乏导致难以对主板的可靠性进行准确地分析。为解决这一问题，本文提出了一种基于故障树的主板可靠性分析方法。首先以某典型笔记本主板为例，为该主板建立了面向主板功能测试环节的故障树，然后针对主板维修数据充足和缺乏的两个场景分别设计了自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法。特别的，为了提升可靠性分析的准确性，本文在建立故障树模型时，充分地考虑了元件间互连信息。实验表明，包含元件互连信息的故障树对主板的可靠性分析更加准确，并且所提方法能够用于测试策略设计，保证了主板测试质量，同时降低了主板测试时间成本。

#### 3.1 主板故障树的建立

##### 3.1.1 普通的主板故障树建立

现有的可靠性分析方法忽略了造成主板测项故障的直接原因，使得其分析结果难以用于测试策略设计过程。为解决上述问题，本章针对笔记本功能测试环节为笔记本主板建立一棵故障树，进一步分析其可靠性。为描述建立故障树的过程，本章以某典型笔记本主板为例，分析引起该主板中各测项发生故障的根本原因。某典型笔记本主板的电路结构如图 3.1 所示。

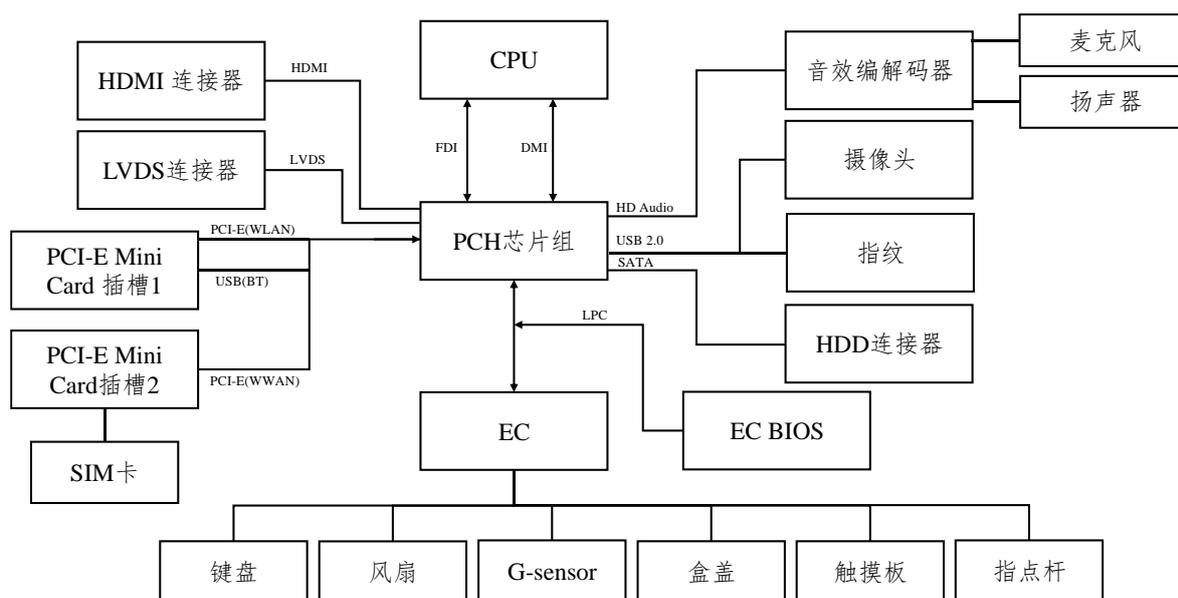


图 3.1 某典型笔记本主板的电路图

Fig. 3.1 The circuit diagram of a typical laptop motherboard

由图 3.1 可知, 该典型笔记本主板包含 16 个功能功能模块, 而主板的功能测项如表 3.1 所示, 因此, 图 3.1 中的功能模块与表 3.1 中的主板功能测项一一对应。在主板功能测试环节, 若某测项测试不通过, 则主板视为故障主板; 反之, 若所有测项均通过测试, 则主板被视为合格主板。

为计算主板和各测项的可靠性, 本章建立了以主板故障为顶事件的故障树。其中, 故障树的中间事件是各测项故障, 底事件是导致各测项发生故障的根本原因。由于在主板功能测试环节, 某个测项故障就会导致主板功能测试不通过, 所以顶事件和中间事件间由或门连接。

为系统地分析导致主板功能测项故障的根本原因, 本文将主板分为四个部分: 嵌入式控制模块(embedded controller module, EC), 中央处理器模块(central processing unit, CPU), 平台控制单元模块(platform controller hub, PCH)和功能模块。其中, EC 模块包括 EC 芯片和 EC BIOS, 所以 EC 芯片故障和 EC BIOS 故障会导致 EC 模块故障; CPU 模块指 CPU 芯片, 所以 CPU 芯片故障会导致该模块故障。上述 EC 芯片故障、CPU 芯片故障和 PCH 芯片组故障均属于芯片故障。由于主板生产工艺中需要将元器件焊接在印刷电路板上, 会存在虚焊等焊接、假焊等问题, 所以芯片故障原因包括两类: 芯片自身故障和芯片焊接故障。

表 3.1 某典型笔记本主板的功能测试项目  
Tab. 3.1 Functional test items for a typical laptop motherboard

编号	测项名称	编号	测项名称
1	无线广域网功能测项	9	LVDS 功能测项
2	键盘功能测项	10	指纹功能测项
3	风扇功能测项	11	摄像头功能测项
4	G-sensor 功能测项	12	面板功能测项
5	盒盖功能测项	13	蓝牙功能测项
6	触摸板功能测项	14	无线局域网功能测项
7	指点杆功能测项	15	扬声器功能测项
8	HDMI 功能测项	16	麦克风功能测项

由图 3.1 可知, 功能模块可分为两个部分: 一部分功能模块直接连接 PCH 模块, 然后 PCH 模块通过 FDI 总线和 DMI 总线连接 CPU 模块, 通过 LPC 总线连接 EC 模块, 这类功能模块包括摄像头模块、指纹模块等; 一部分功能模块通过 EC 模块连接 PCH 模块, 然后 PCH 模块通过 FDI 总线和 DMI 总线连接 CPU 模块, 这类功能模块包括键盘

模块、风扇模块。因此，导致 PCH 模块故障的原因可分为两类，分别称为 PCH 模块故障 1 和 PCH 模块故障 2。PCH 模块故障 1 由 CPU 模块故障和 EC 模块故障引起，PCH 模块故障 2 由 CPU 模块故障引起。

导致功能模块发生故障的原因可以分为两部分：功能模块本身故障以及与该功能模块直接相连的模块故障。例如，导致图 3.1 中的键盘功能测项发生故障的原因可以分为两部分，分别是键盘本身故障和 EC 模块故障。

根据上述过程分析导致主板各功能模块发生故障的原因，确定了故障树的中间事件如表 3.2 所示。由表 3.2 可知，这颗故障树包含 22 个中间事件，其中，中间事件  $M_1$  到  $M_{16}$  与主板的测项（其编号为 1 到 16）一一对应，中间事件  $M_{17}$  到  $M_{22}$  是导致功能测项故障的直接原因。

表 3.2 笔记本主板故障树的中间事件描述

Tab. 3.2 The description of intermediate events in the fault tree of a laptop motherboard

符号	故障描述	符号	故障描述
$M_1$	无线广域网故障	$M_{12}$	面板故障
$M_2$	键盘故障	$M_{13}$	蓝牙故障
$M_3$	风扇故障	$M_{14}$	无线局域网故障
$M_4$	G-sensor 故障	$M_{15}$	扬声器故障
$M_5$	盒盖故障	$M_{16}$	麦克风故障
$M_6$	触摸板故障	$M_{17}$	PCH 芯片组模块 1 故障
$M_7$	指点杆故障	$M_{18}$	PCH 芯片组模块 2 故障
$M_8$	HDMI 故障	$M_{19}$	PCH 芯片组自身故障
$M_9$	LVDS 故障	$M_{20}$	CPU 模块故障
$M_{10}$	指纹故障	$M_{21}$	EC 模块故障
$M_{11}$	摄像头故障	$M_{22}$	EC 芯片故障

接下来，结合主板维修记录和主板电路结构，进一步分析导致中间事件  $M_{17}$ 、 $M_{18}$ 、 $M_{19}$ 、 $M_{20}$ 、 $M_{21}$  和  $M_{22}$  发生故障的根本原因，并将这些原因作为故障树的底事件。某典型笔记本主板故障树的底事件如表 3.3 所示。

下面以键盘功能测项不良为例，详细介绍以键盘功能测项不良为顶事件的故障树分支的建立过程。主要包含两个步骤：首先根据主板的电路结构将主板分成 4 个模块，分析哪些模块造成该功能测项发生故障，这是造成键盘功能测项不良的直接原因，这些模块故障作为故障树分支的中间事件，再逐一分析造成模块故障的根本原因，作为故障树

分支的底事件。

**表 3.3 笔记本主板故障树的底事件描述**  
**Tab. 3.3 Bottom event description of the laptop motherboard fault tree**

符号	故障描述	符号	故障描述
X <sub>1</sub>	读卡器故障	X <sub>13</sub>	蓝牙自身模块故障
X <sub>2</sub>	PCH 芯片焊接不良	X <sub>14</sub>	风扇自身故障
X <sub>3</sub>	PCH 芯片损坏	X <sub>15</sub>	G-Sensor 自身故障
X <sub>4</sub>	CPU 芯片焊接不良	X <sub>16</sub>	盒盖自身故障
X <sub>5</sub>	CPU 芯片损坏	X <sub>17</sub>	触摸板自身故障
X <sub>6</sub>	EC BIOS 故障	X <sub>18</sub>	指点杆自身故障
X <sub>7</sub>	EC 芯片焊接不良	X <sub>19</sub>	LVDS 自身故障
X <sub>8</sub>	EC 芯片损坏	X <sub>20</sub>	摄像头自身故障
X <sub>9</sub>	键盘自身模块故障	X <sub>21</sub>	扬声器自身故障
X <sub>10</sub>	HDMI 连接器故障	X <sub>22</sub>	麦克风自身故障
X <sub>11</sub>	指纹模块自身故障	X <sub>23</sub>	音效编解码器自身故障
X <sub>12</sub>	HDD 模块故障	X <sub>24</sub>	无线局域网自身故障

键盘通过 EC 模块与外界相连，EC 模块通过 LPC 总线和 PCH 芯片组相连，所以中间事件“键盘故障”通过或门与键盘自身模块故障（X<sub>9</sub>）、EC 模块故障（M<sub>21</sub>）和 PCH 芯片组模块故障 2（M<sub>18</sub>）相连。其中，键盘自身模块故障是底事件，无需再往下分析其故障原因。EC 模块故障可能由 EC 芯片故障（X<sub>8</sub>）或 EC 芯片焊接不良（X<sub>7</sub>）引起，所以 EC 模块故障通过或门连接 EC 芯片故障和 EC 芯片焊接不良。在 PCH 芯片组模块 2 中，PCH 芯片组仅通过 FDI 总线和 DMI 总线和 CPU 芯片连接，所以中间事件“PCH 芯片组模块故障 2”通过或门与 PCH 芯片组故障（M<sub>19</sub>）和 CPU 模块故障（M<sub>20</sub>）相连。以“键盘故障”为顶事件的故障树分支如图 3.2 所示：

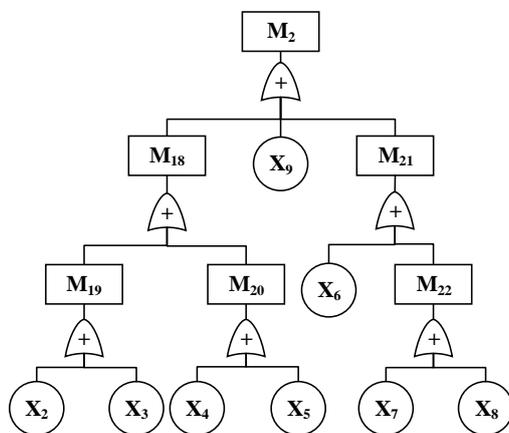


图 3.2 键盘功能模块的故障树分支

Fig. 3.2 The fault tree branching for the keyboard function module

按照上述过程，利用主板电路结构，逐一分析其他功能测项的故障原因，并建立以其他功能测项不良为顶事件的故障树分支。图 3.1 中的某典型笔记本主板包含 16 个功能测项，其中任意一个功能测项发生故障都将判定主板不合格，即主板发生故障。因此，功能测项发生故障构成主板故障树的中间事件，以此建立针对主板功能测试阶段的主板故障树，如图 3.3 所示。

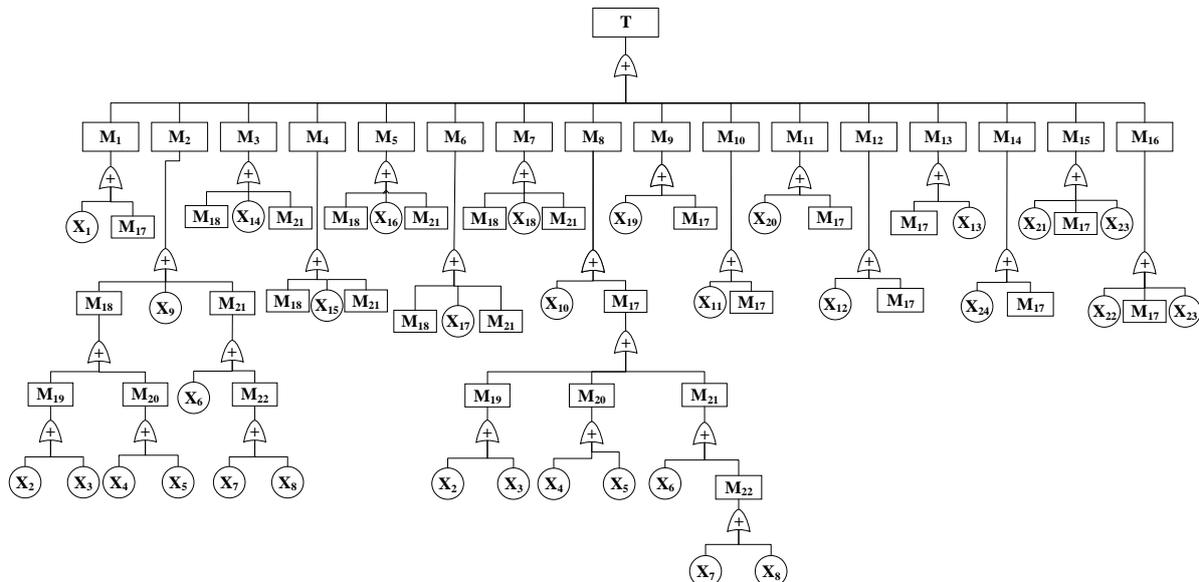


图 3.3 某典型笔记本主板的故障树

Fig. 3.3 The fault tree of a typical laptop motherboard

其中，T 代表故障树的顶事件“笔记本主板故障”，M<sub>1</sub> 到 M<sub>16</sub> 分别代表功能测项发生故障，并且 M<sub>1</sub> 到 M<sub>16</sub> 通过或门连接顶事件，它是造成主板故障的直接原因。M<sub>17</sub> 和 M<sub>18</sub> 分别代表中间事件“PCH 芯片组模块故障 1”和中间事件“PCH 芯片组模块故障 2”。X<sub>i</sub> (i=1,2,...,24) 代表底事件，它是造成其上层事件发生故障的根本原因。为使故障树更加清晰，图中省略了部分中间事件的故障树分支。例如，以中间事件 M<sub>17</sub>、M<sub>18</sub> 和 M<sub>21</sub> 为顶事件的故障树分支仅出现一次。

每颗故障树分支均对应一个数学表达式，称为布尔表达式。假设底事件 X<sub>i</sub> 的发生概率是 P<sub>X<sub>i</sub></sub> (i=1,2,...,24)，根据公式(2.1-2.2)和图 3.3 中的故障树结构得到故障树的中间事件发生概率的布尔表达式，如下所示：

$$\begin{aligned}
 P_{M_j} &= P_{\bigcup_{X_i \in S_j} X_i} \\
 &= 1 - \prod_{X_i \in S_j} (1 - P_{X_i}), \\
 &i = 1, 2, \dots, 24, \\
 &j = 1, 2, \dots, 16,
 \end{aligned}
 \tag{3.1}$$

其中，S<sub>j</sub>是故障树中各中间事件所连接的底事件的集合，这些底事件导致中间事件发生。

中间事件  $M_j$  ( $j=1,2,\dots,24$ ) 所连接的底事件集合如表 3.4 所示:

**表 3.4 某典型笔记本主板中各测项的故障原因集合**  
**Tab. 3.4 Collections of failure causes for each test item in a typical laptop motherboard**

符号	故障描述	底事件集合
$M_1$	无线广域网故障	$\{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8\}$
$M_2$	键盘故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9\}$
$M_3$	风扇故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{14}\}$
$M_4$	G-sensor 故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{15}\}$
$M_5$	盒盖故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{16}\}$
$M_6$	触摸板故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{17}\}$
$M_7$	指点杆故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{18}\}$
$M_8$	HDMI 故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{10}\}$
$M_9$	LVDS 故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{19}\}$
$M_{10}$	指纹故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{11}\}$
$M_{11}$	摄像头故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{20}\}$
$M_{12}$	面板故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{12}\}$
$M_{13}$	蓝牙故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{13}\}$
$M_{14}$	无线局域网故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{24}\}$
$M_{15}$	扬声器故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{21}, X_{23}\}$
$M_{16}$	麦克风故障	$\{X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_{22}, X_{23}\}$

### 3.1.2 包含元件互连信息的主板故障树建立

针对现有的主板可靠性分析相关研究大多忽略了元件间互连故障的问题, 本节充分考虑元件间的互连信息, 为笔记本主板建立包含元件互连信息的故障树。

依然以图 3.1 中的某典型笔记本主板为例, 依然将主板结构分成 4 个模块, 从主板测项出发逐一分析造成主板故障的根本原因。下面简要描述该笔记本主板的包含元件互连信息的故障树的建立过程。

在 3.1.1 节的基础上, 充分考虑主板原件间的连接信息, 所以可以从三方面分析导致功能测项不良的原因: 功能模块自身不良, 与功能模块直接相连的模块不良以及连接两模块的总线不良。

逐个分析造成表 3.2 里中间事件  $M_{17}$ 、 $M_{18}$ 、 $M_{19}$ 、 $M_{20}$ 、 $M_{21}$  和  $M_{22}$  发生的根本原因, 这些原因构成故障树的底事件, 如表 3.5 所示。相比于 3.1.1 节所建立的故障树, 该故障树的底事件更加详尽, 由 23 个增加到 32 个。

接下来，同样以键盘功能测项不良为例，过程中充分考虑元件间的互连信息，分析造成该测项不良的原因，然后建立以键盘功能测项故障为顶事件的故障树分支。下面详细介绍其故障树分支建立过程。

表 3.5 包含元件互连信息的笔记本主板故障树的底事件描述

Tab. 3.5 Bottom event description of the laptop motherboard fault tree containing component interconnect information

符号	故障描述	符号	故障描述
X <sub>1</sub>	读卡器故障	X <sub>17</sub>	USB2.0 接口故障
X <sub>2</sub>	PCI-E 总线故障	X <sub>18</sub>	HDD 模块故障
X <sub>3</sub>	PCH 芯片焊接不良	X <sub>19</sub>	SATA 总线故障
X <sub>4</sub>	PCH 芯片损坏	X <sub>20</sub>	蓝牙自身模块故障
X <sub>5</sub>	CPU 芯片焊接不良	X <sub>21</sub>	风扇自身故障
X <sub>6</sub>	CPU 芯片损坏	X <sub>22</sub>	G-Sensor 自身故障
X <sub>7</sub>	FDI 总线故障	X <sub>23</sub>	盒盖自身故障
X <sub>8</sub>	DMI 总线故障	X <sub>24</sub>	触摸板自身故障
X <sub>9</sub>	EC BIOS 故障	X <sub>25</sub>	指点杆自身故障
X <sub>10</sub>	LPC 总线故障	X <sub>26</sub>	LVDS 连接器自身故障
X <sub>11</sub>	EC 芯片焊接不良	X <sub>27</sub>	LVDS 自身故障
X <sub>12</sub>	EC 芯片损坏	X <sub>28</sub>	摄像头自身故障
X <sub>13</sub>	键盘自身模块故障	X <sub>29</sub>	扬声器自身故障
X <sub>14</sub>	HDMI 连接器故障	X <sub>30</sub>	麦克风自身故障
X <sub>15</sub>	HDMI 接口故障	X <sub>31</sub>	音效编解码器自身故障
X <sub>16</sub>	指纹模块自身故障	X <sub>32</sub>	HD Audio 故障

键盘通过 EC 模块与外界相连，EC 模块通过 LPC 总线和 PCH 芯片组模块相连，所以键盘故障 (M<sub>2</sub>) 通过或门与键盘自身模块故障 (X<sub>13</sub>)、LPC 总线故障 (X<sub>10</sub>)、PCH 芯片组模块故障 2 (M<sub>18</sub>) 相连。以“键盘故障”为顶事件的故障树分支如图 3.4 所示：

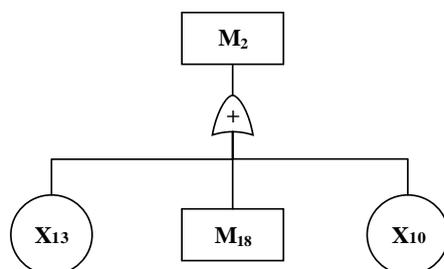


图 3.4 以键盘功能测项故障为顶事件的故障树分支

Fig. 3.4 Fault tree branching with keyboard functional test item failure as top event

其中，键盘自身模块故障和 LPC 总线故障均是底事件，无需再往下分析其故障原因。

在 PCH 芯片组模块中，PCH 芯片组仅通过 FDI 总线和 DMI 总线和 CPU 芯片连接，所以中间事件“PCH 芯片组模块 2 故障”通过或门与 FDI 总线故障 ( $X_7$ )、DMI 总线故障 ( $X_8$ )、PCH 芯片组故障 ( $M_{19}$ ) 和 CPU 模块故障 ( $M_{20}$ ) 相连。

根据笔记本主板的电路图逐步分析其他测项的故障原因，最终确定笔记本主板的故障树如图 3.5 所示。其中， $T$ 代表故障树的顶事件“笔记本主板故障”， $M_{17}$ 和 $M_{18}$ 分别代表中间事件“PCH 芯片组模块 1 故障”和“PCH 芯片组模块 2 故障”，图中省略了部分中间事件 $M_{17}$ 和 $M_{18}$ 的故障树分支。

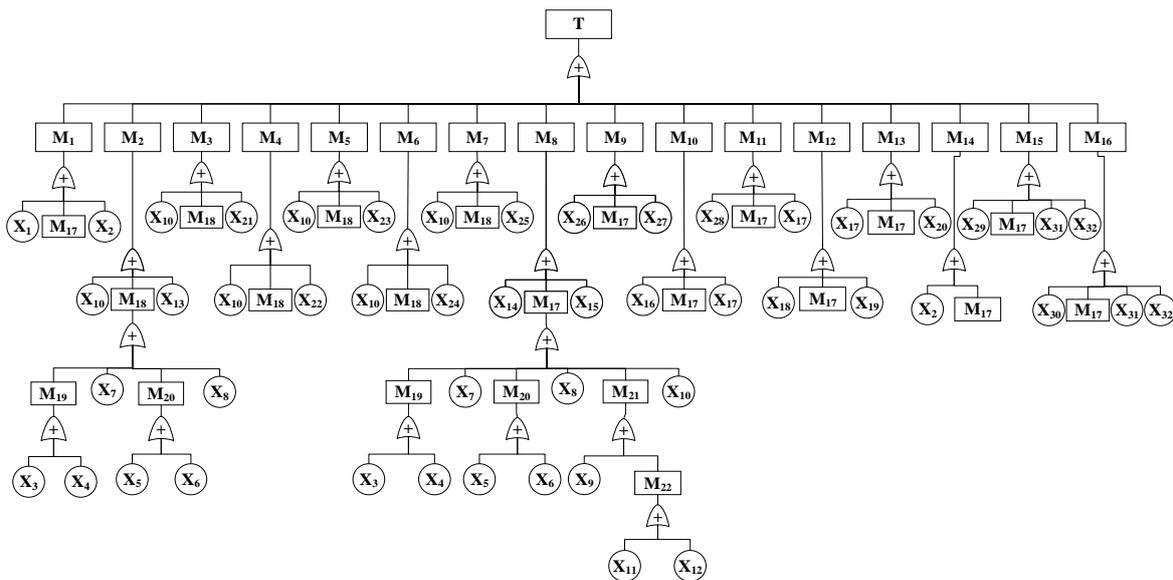


图 3.5 笔记本主板的故障树(包含元件互连信息)

Fig. 3.5 Fault tree of a laptop motherboard (with component interconnection information)

由于故障树结构和布尔表达式一一对应，因此可以根据布尔表达式的化简公式将笔记本主板的故障树化简，化简后得到以测项故障为中间事件的故障树分支所对应的简洁布尔表达式。

化简后，中间事件“测项故障”均通过或门连接各底事件，利用公式(2.2)计算得到各测项故障的发生概率表达式。假设底事件 $X_i$ 的发生概率为 $P_{X_i}$ ，则某中间事件 ( $M_j$ ) 对应的布尔表达式和发生概率表达式如下所示：

$$\begin{aligned}
 P_{M_j} &= P_{\cup_{i \in S_j} X_i} \\
 &= 1 - \prod_{i \in S_j} (1 - P_{X_i})
 \end{aligned}
 \tag{3.2}$$

$$i=1,2,\dots,32, j=1,2,\dots,16$$

其中， $S_j$ 是以测项故障 $M_j$ 为顶事件的故障树分支中底事件所构成的集合。集合中的具体

内容如表 3.6 所示。

**表 3.6 某典型笔记本主板中各测项的故障原因集合(包含元件互连信息)**  
**Tab. 3.6 Failure cause sets for each test in a typical laptop motherboard (with component interconnect information)**

符号	故障描述	底事件集合
M <sub>1</sub>	无线广域网故障	{X <sub>1</sub> ,X <sub>2</sub> ,X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> }
M <sub>2</sub>	键盘故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>10</sub> ,X <sub>13</sub> }
M <sub>3</sub>	风扇故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>10</sub> ,X <sub>21</sub> }
M <sub>4</sub>	G-sensor 故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>10</sub> ,X <sub>22</sub> }
M <sub>5</sub>	盒盖故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>10</sub> ,X <sub>23</sub> }
M <sub>6</sub>	触摸板故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>10</sub> ,X <sub>24</sub> }
M <sub>7</sub>	指点杆故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>10</sub> ,X <sub>25</sub> }
M <sub>8</sub>	HDMI 故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>14</sub> ,X <sub>15</sub> }
M <sub>9</sub>	LVDS 故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>26</sub> ,X <sub>27</sub> }
M <sub>10</sub>	指纹故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>16</sub> ,X <sub>17</sub> }
M <sub>11</sub>	摄像头故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>28</sub> ,X <sub>17</sub> }
M <sub>12</sub>	面板故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>18</sub> ,X <sub>19</sub> }
M <sub>13</sub>	蓝牙故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>17</sub> ,X <sub>20</sub> }
M <sub>14</sub>	无线局域网故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> }
M <sub>15</sub>	扬声器故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>29</sub> ,X <sub>31</sub> ,X <sub>32</sub> }
M <sub>16</sub>	麦克风故障	{X <sub>3</sub> ,X <sub>4</sub> ,X <sub>5</sub> ,X <sub>6</sub> ,X <sub>7</sub> ,X <sub>8</sub> ,X <sub>9</sub> ,X <sub>10</sub> ,X <sub>11</sub> ,X <sub>12</sub> ,X <sub>30</sub> ,X <sub>31</sub> ,X <sub>32</sub> }

## 3.2 主板可靠性分析方法

### 3.2.1 自底向上的主板可靠性分析方法

我们将根据底事件的发生概率计算中间事件和顶事件的发生概率的方法称为自顶向上的主板可靠性分析方法。自底向上的主板可靠性分析方法包含两个步骤：首先利用主板的维修数据获得各基本事件发生的概率，然后利用笔记本主板的故障树结构信息求得中间事件和顶事件的发生概率。具体过程如下：

本文对主板功能测试过程中的返修记录进行统计分析，得到引发主板不良的各种故障因素的发生频率，将上述频率作为基本事件 $X_i$ 的发生概率，记为 $[P_{min}(X_i), P_{max}(X_i)]$ ，其中 $i = 1, \dots, 32$ 。

由图 3.5 中故障树的结构可知，故障树中的底事件均独立，下面通过公式(2.2)和表 3.6 中的数据定量计算中间事件的发生概率，记为 $[P_{min}(M_j), P_{max}(M_j)]$ 。本文将良率作

为衡量主板可靠性的指标，因此主板中各测项的良率范围如公式(3.3)所示：

$$\begin{aligned} \beta_{min}(M_j) &= 1 - P_{max}(M_j), \\ \beta_{max}(M_j) &= 1 - P_{min}(M_j), \end{aligned} \tag{3.3}$$

其中， $j = 1, \dots, 16$ ，表示中间事件的序号。

### 3.2.2 自顶向下的主板可靠性分析方法

目前主流的笔记本制造厂商对于某一型号的笔记本进行功能测试时往往需要1年左右的时间，而返修记录的积累可能需要数月，才能够较为准确地统计出基本事件的发生概率。所以，自底向上的主板可靠性分析方法仅在返修数据充足的前提下有效。为应对主板测试初期返修数据不足的情况下，本文提出了自顶向下的主板可靠性分析方法，该方法通过顶事件的发生概率反推基本事件和中间事件的发生概率。

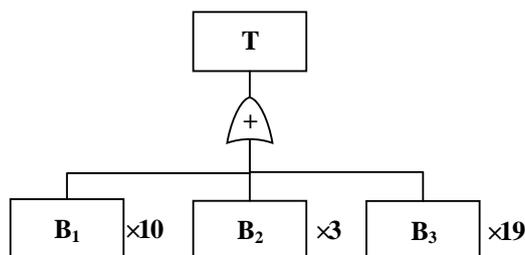
自顶向下的阈值设定方法包含两个主要步骤：将基本事件依据故障类型进行分类、利用概率公式反推中间事件和顶事件概率。

将基本事件依据故障类型分为三类：总线及接口连接故障、芯片焊接不良以及原件或芯片自身故障，三类分别记为： $B_1$ 、 $B_2$ 和 $B_3$ 。则基本事件分类结果如表 3.7 所示。

**表 3.7 基本事件分类结果**  
**Tab. 3.7 The results of basic event classification**

类别	包含的基本事件
$B_1$	$X_2, X_7, X_8, X_9, X_{10}, X_{15}, X_{17}, X_{19}, X_{27}, X_{32}$
$B_2$	$X_3, X_5, X_{11}$
$B_3$	$X_1, X_4, X_6, X_{12}, X_{13}, X_{14}, X_{16}, X_{18}, X_{20}, X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}, X_{28}, X_{29}, X_{30}, X_{31}$

根据分类后的基本事件对图 3.5 中故障树做进一步的化简处理，该图中有 10 个基本事件属于 $B_1$ 类，有 3 个基本事件属于 $B_2$ 类，有 19 个基本事件属于 $B_3$ 类。因此分类后的笔记本主板的故障树如图 3.6 所示。



**图 3.6 底事件分类后的笔记本主板的故障树**

**Fig. 3.6 The fault tree of a laptop motherboard after bottom event classification**

依据专家经验得知，同类基本事件的发生概率接近，而不同类基本事件的发生概率

存在差别,因此假定同类基本事件的发生概率相同,不同类基本事件的发生概率成比例,其比例根据实际生产经验取值。此处将 $B_1$ 、 $B_2$ 、 $B_3$ 三类基本事件发生的概率之比记为: $1:r_1:r_2$ ,则可以将三类基本事件发生的概率分别表示为 $p$ 、 $r_1p$ 、 $r_2p$ 。根据图 3.6 将顶事件发生概率 $P_T$ 表示为关于 $p$ 、 $r_1$ 和 $r_2$ 的函数:

$$P_T = 1 - (1 - p)^{10} \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^{19} \quad (3.4)$$

根据公式(3.4)和表 3.7 中集合的内容计算出中间事件的发生概率,即测项不良率关于 $p$ 、 $r_1$ 和 $r_2$ 的表达式如表 3.8 所示。

表 3.8 基本事件发生概率成比例时各测项的不良率表达式

Tab. 3.8 The expression for the failure rate of each test item when the occurrence probability of the basic event is proportional

编号	功能测项	不良率表达式
1	无线广域网功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
2	键盘功能测项	$1 - (1 - p)^3 \cdot (1 - r_1p)^2 \cdot (1 - r_2p)^3$
3	风扇功能测项	$1 - (1 - p)^3 \cdot (1 - r_1p)^2 \cdot (1 - r_2p)^3$
4	G-sensor 功能测项	$1 - (1 - p)^3 \cdot (1 - r_1p)^2 \cdot (1 - r_2p)^3$
5	盒盖功能测项	$1 - (1 - p)^3 \cdot (1 - r_1p)^2 \cdot (1 - r_2p)^3$
6	触摸板功能测项	$1 - (1 - p)^3 \cdot (1 - r_1p)^2 \cdot (1 - r_2p)^3$
7	指点杆功能测项	$1 - (1 - p)^3 \cdot (1 - r_1p)^2 \cdot (1 - r_2p)^3$
8	HDMI 功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
9	LVDS 功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
10	指纹功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
11	摄像头功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
12	面板功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
13	蓝牙功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^4$
14	无线局域网功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^3$
15	扬声器功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^5$
16	麦克风功能测项	$1 - (1 - p)^5 \cdot (1 - r_1p)^3 \cdot (1 - r_2p)^5$

根据专家经验为 $r_1$ 和 $r_2$ 设定合适的取值,根据主板测试结果或工厂要求的可靠性指标确定 $P_T$ 的范围,再结合公式(3.4)即可求出 $p$ 的范围。将结果代入表 3.8 中各测项的不良率表达式,从而获得各测项发生故障的概率范围,记为 $[P_{min}(M_j), P_{max}(M_j)]$ ,其中 $j = 1, \dots, 16$ 。进一步地,利用公式(3.3)获得主板中各测项的可靠性范围。

### 3.3 基于可靠性分析的主板功能测试策略设计

#### 3.3.1 整体流程

为验证将主板可靠性分析用于主板功能测试阶段的可行性，该节将主板可靠性分析结果用于该阶段的主板功能测试策略设计。首先在第 3.1.2 节所建立的包含元件互连信息的故障树的基础上，分别使用自顶向下和自底向上的主板可靠性分析方法，获得主板和测项的可靠性指标范围，然后通过设计测试策略将测项的良率控制在可靠性范围之内。

将主板可靠性分析结果用于该阶段的主板功能测试策略设计主要包含两个部分，分别是测项的可靠性指标设定和主板功能测试策略设计，其整体流程如图 4.1 所示：

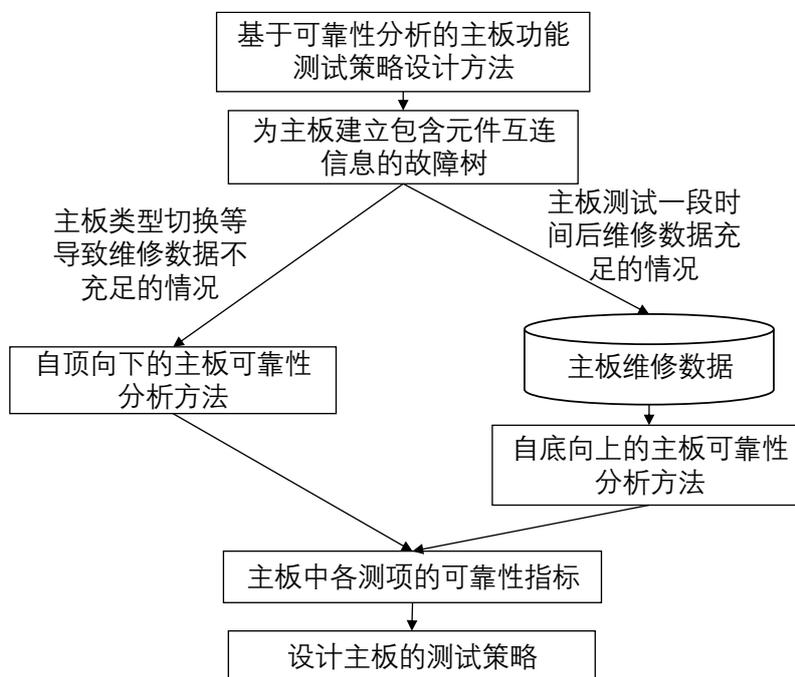


图 3.7 测试策略设计过程

Fig. 3.7 Flow of functional test strategy design method based on reliability analysis

#### 3.3.2 测项的可靠性指标设定

在主板测试初期维修数据不充足的情况下，使用自顶向下的可靠性分析方法，得到主板中所有测项的可靠性范围；在维修数据积累到一定量时，使用自底向上的可靠性分析方法得到测项的可靠性范围。

本文将良率作为衡量主板可靠性的指标，所以为各测项设定一个良率阈值，然后将测项的良率控制在对应阈值之下，即可保证主板的可靠性。由于测项的良率阈值在良率范围内波动，所以测项的良率阈值设定范围如公式(3.5)所示。其中， $[t_{min}(M_j), t_{max}(M_j)]$

和 $[\beta_{min}(M_j), \beta_{max}(M_j)]$ 分别表示测项 $M_j$ 的良率阈值设定范围和良率范围,  $j = 1, 2, \dots, m$ 。

$\delta(M_j)$ 是根据专家经验设定的经验值, 当 $\delta(M_j) \geq 0$ 时, 良率阈值设定较高, 对测项质量的把控相对严格; 反之, 对测项质量的要求相对放松。将 $[t_{min}(M_j), t_{max}(M_j)]$ 范围内的平均值作为测项的良率阈值, 记为 $t(M_j)$ 。

$$\begin{aligned} t_{min}(M_j) &= \beta_{min}(M_j) + \delta(M_j), \\ t_{max}(M_j) &= \beta_{max}(M_j) + \delta(M_j), \end{aligned} \quad (3.5)$$

### 3.3.3 主板功能测试策略设计

在主板功能测试过程中, 时刻关注实际产线上主板中各测项的良率, 将其与对应的阈值 $t(M_j)$ 作比较, 即可得到主板的测试策略, 如公式(3.6)所示。

$$\mathbf{s} = [s(M_1), s(M_2), \dots, s(M_j)], j = 1, \dots, m, \quad (3.6)$$

其中,  $s(M_j) \in \mathbb{Z}_2 = \{0, 1\}$ 表示测项 $M_j$ 测试与否的决策, 其取值为0和1, 分别表示对该测项进行测试和不测试, 由如下公式确定:

$$s(M_j) = \neg(y(M_j) \geq t(M_j)), j = 1, \dots, m, \quad (3.7)$$

其中,  $y(M_j)$ 表示主板功能测试过程中测项 $M_j$ 的实际良率,  $t(M_j)$ 表示测项 $M_j$ 的良率阈值。为保证主板的可靠性, 当某测项良率未达到良率阈值时, 需要对该测项进行测试; 反之, 不需要测试。由于工厂中实际生产环境复杂, 主板或测项的良率会随主板测试结果发生变化, 为避免测项 $M_j$ 测试状态的频繁变动, 需在公式(3.5)中设定合适的参数 $\delta(M_j)$ 。

## 3.4 数据集及评价指标

### 3.4.1 笔记本主板数据集

在工厂中采集到一批某典型笔记本主板的测试结果, 并统计出各测项的测试时间。将主板的测试结果记为 $\mathbf{D} \in \mathbb{Z}_2^{n_{pcb} \times m}$ , 其中 $\mathbb{Z}_2 \in \{0, 1\}$ , “0”代表测试结果为不通过, “1”表示通过测试,  $n_{pcb}$ 为主板数目,  $m$ 为每块主板所包括的测试项目总数。各测项的测试时间存放到矩阵 $\mathbf{T} \in \mathbb{R}^{n_{pcb} \times m}$ 中, 其中,  $\mathbb{R}$ 表示实数。

首先对主板测试结果进行分析，良率是指主板中某测项的测试结果里“1”元素出现的概率，第  $j$  个测项的良率  $\mu_j$  计算如下。其中， $D_{i,j}$  表示第  $i$  块主板的第  $j$  个测项的测试结果，则这  $m$  个测项的良率构成向量  $\boldsymbol{\mu} = [\mu_1, \mu_2, \dots, \mu_m]^T$ 。

$$\mu_j = \frac{\sum_{i=1}^n D_{i,j}}{n_{pcb}}, j = 1, 2, \dots, m \quad (3.8)$$

接着，统计出这一批主板中各测项的平均测试时间，如公式(3.9)所示：

$$t_j = \frac{\sum_{i=1}^n T_{i,j}}{n_{pcb}}, j = 1, 2, \dots, m, \quad (3.9)$$

其中， $t_j$  表示第  $j$  个测项的平均测试时间， $T_{i,j}$  表示第  $i$  块主板的第  $j$  个测项的测试时间。所有测项的平均测试时间构成长度为  $m$  的向量  $\boldsymbol{t} = [t_1, t_2, \dots, t_m]^T$ 。

具体的，工厂中的这批数据包含 1715 块主板的测试结果，每块主板包含 16 个测试项目，因此公式(3.8-3.9)中  $n_{pcb} = 1715$ ， $m=16$ 。

通过上述过程计算出主板测项的不良率，并统计出主板的平均测试时间。某典型笔记本主板中各测项的真实良率和平均测试时间的具体情况如表 3.9 所示。

表 3.9 某典型笔记本主板中各测项的真实良率和平均测试时间

Tab. 3.9 The true yield and average test time for each test item in a typical laptop motherboard

符号	测试项目	不良率/%	平均测试时间/s
M <sub>1</sub>	无线广域网功能测项	0.0287	2.44
M <sub>2</sub>	键盘功能测项	0.0191	5.21
M <sub>3</sub>	风扇功能测项	0.0190	0.44
M <sub>4</sub>	G-sensor 功能测项	0.0191	0.33
M <sub>5</sub>	盒盖功能测项	0.0188	10.89
M <sub>6</sub>	触摸板功能测项	0.0190	1.85
M <sub>7</sub>	指点杆功能测项	0.0188	0.68
M <sub>8</sub>	HDMI 功能测项	0.0287	7.73
M <sub>9</sub>	LVDS 功能测项	0.0289	9.34
M <sub>10</sub>	指纹功能测项	0.0286	8.78
M <sub>11</sub>	摄像头功能测项	0.0285	1.35
M <sub>12</sub>	面板功能测项	0.0288	1.41
M <sub>13</sub>	蓝牙功能测项	0.0286	1.41
M <sub>14</sub>	无线局域网功能测项	0.0264	62.93
M <sub>15</sub>	扬声器功能测项	0.0306	0.82
M <sub>16</sub>	麦克风功能测项	0.0305	0.72

由于目前可用的实际数据偏少，需要从实际数据中提取除出各测项的特征，然后根

据特征合成仿真数据。

本文生成仿真数据的方式是将表 3.9 中 16 个测试项目的不良率作为仿真数据中主板各测项的不良率，其中，不良率是指主板中某测项的测试结果里“0”元素出现的概率。通过这种方法控制了仿真数据中测项的测试结果中“0”元素出现的次数，保证了各测项的不良率特征不变，据此生成 50000 块主板的测试结果，这些测试结果组成二值矩阵  $\mathbf{D} \in \mathbb{Z}_2^{50000 \times 16}$ ，矩阵由元素“0”和“1”构成。

### 3.4.2 测试策略的评价指标

为衡量测试策略在主板功能测试环节的测试效率和测试质量，本文使用了三个评判指标：平均测试时间、缺陷水平和平均测试成本<sup>[5,31,68]</sup>。其中，平均测试时间是指当一批主板使用某测试策略进行功能测试时平均每块主板所花费的测试时间，该指标能够很好的反应测试策略对应的测试效率；缺陷水平是指经过测试策略测试的一批主板中仍包含的缺陷主板的比例，该指标能够衡量测试策略对应的测试质量；平均测试总成本是指平均每块主板所花费的总时间，该时间由漏检的不良主板所造成的维修时间和主板的平均测试时间共同构成。

假设有  $n$  块主板，主板中各测项的测试时间为  $\mathbf{t} = [t_1, t_2, \dots, t_i]^T$ ，其中  $t_i$  表示第  $i$  个测项的平均测试时间。使用测试策略  $\mathbf{s} = [s_1, s_2, \dots, s_i]^T$  对其进行功能测试，其中  $s_i \in \mathbb{Z}_2 = \{0,1\}$  表示测项  $M_j$  测试与否的决策，其取值为 0 和 1，分别表示对该测项进行测试和不测试。测试策略  $\mathbf{s}$  对应的平均测试时间、缺陷水平和平均测试总成本计算如下。

使用测试策略  $\mathbf{s}$  时，每块主板的平均测试时间  $f_1$  计算如下：

$$f_1 = \mathbf{t}^T \mathbf{s} \quad (3.10)$$

缺陷水平是指假阳性主板占有所有测试通过的主板的比例，其中假阳性主板指虽通过测试，本质上却是不良的主板。分别利用公式 (3.11) 和 (3.12) 计算使用测试策略  $\mathbf{s}$  时测试通过的主板数目  $p$  和真实的合格主板数目  $p_0$ ，然后利用公式 (3.13) 计算缺陷水平  $f_2$ ：

$$p = n - \sum_{k=1}^n \left[ \left( \bigvee_{i=1}^m (\mathbf{D}_{i,k} = 0 | s_i = 1) \right) \equiv \top \right], \quad (3.11)$$

$$p_0 = n - \sum_{k=1}^n \left[ \left( \bigvee_{i=1}^m (\mathbf{D}_{i,k} = 0) \right) \equiv \top \right], \quad (3.12)$$

$$f_2 = \frac{p-p_0}{p} \quad (3.13)$$

测试总成本为测试时间和假阳性主板造成的维修成本的总和，该指标反应了测试策

略的综合水平，则平均测试总成本的计算公式如下：

$$f = f_1 + \frac{w \cdot p \cdot f_2}{n}, \quad (3.14)$$

其中， $w$ 为一块假阳性主板所造成的维修成本。

## 3.5 实验与分析

### 3.5.1 数据准备

由于厂商需要数月积累返修数据，所以需要将 1715 块主板的测试结果按 2: 3 划分成  $D_{init}$  和  $D_{steady}$ 。 $D_{init}$  表示某型号主板开始测试阶段获得的测试结果，称为初始阶段，这段时间也是返修数据不断积累的阶段，返修记录可以用来分析主板故障原因并计算故障的发生概率； $D_{steady}$  表示测试稳步进行时所获得的测试结果，该阶段称为稳定阶段。

首先利用公式(3.9)计算实际数据中测试各测项所花费的平均时间，将其作为仿真数据中各测试项目的测试时间，记为  $t$ 。接着分别统计出  $D_{init}$  和  $D_{steady}$  数据中的各测项的良率，即测试结果中“1”所占的比例，实际数据中各测项的测试时间与良率如表 3.9 所示。

表 3.10 实际数据  $D_{init}$  和上  $D_{steady}$  各测项的不良率和平均测试时间

Tab. 3.10 The failure rate and the average test time of each test item on  $D_{init}$  and  $D_{steady}$

符号	功能测项	测试时间/s	$D_{init}$ 不良率/%	$D_{steady}$ 不良率/%
1	无线广域网功能	2.44	0.0167	0.0257
2	键盘功能测项	5.21	0.0189	0.0279
3	风扇功能测项	0.44	0.0201	0.0291
4	G-sensor 功能测	0.33	0.0225	0.0315
5	盒盖功能测项	10.89	0.0181	0.0271
6	触摸板功能测项	1.85	0.0222	0.0312
7	指点杆功能测项	0.68	0.0132	0.0222
8	HDMI 功能测项	7.73	0.0117	0.0207
9	LVDS 功能测项	9.34	0.0233	0.0323
10	指纹功能测项	8.78	0.0152	0.0242
11	摄像头功能测项	1.35	0.0204	0.0294
12	面板功能测项	1.41	0.0196	0.0286
13	蓝牙功能测项	1.41	0.0125	0.0215
14	无线局域网功能	62.93	0.0296	0.0386
15	扬声器功能测项	0.82	0.0293	0.0383
16	麦克风功能测项	0.72	0.0306	0.0396

将实际数据 $D_{init}$ 和 $D_{steady}$ 上各测项的良率作为仿真数据中初始阶段和稳定阶段各测项的良率，即仿真数据中该测项的测试结果里“1”元素出现的概率，据此生成 20000 和 30000 块主板的测试结果，分别记为 $D'_{init} \in \mathbb{Z}_2^{20000 \times 16}$ 和  $D'_{steady} \in \mathbb{Z}_2^{30000 \times 16}$ 。 $D'_{init}$ 和 $D'_{steady}$ 共同组成仿真测试数据 $D'$ 。

将 $D'_{init}$ 和 $D'_{steady}$ 分别按 1: 4 划分成两部分，分别记作 $D'_{init1}$ 、 $D'_{init2}$ 、 $D'_{steady1}$ 和 $D'_{steady2}$ 。将 $D'_{init2}$ 和 $D'_{steady2}$ 作为测试集，在测试集上使用所提测试策略设计方法，通过观察测试集上的测试效果验证所提方法的有效性。

由于测试集上的良率未知，需要从数据集 $D'_{init1}$ 和 $D'_{steady1}$ 上提取测项的良率信息，将 $D'_{init1}$ 和 $D'_{steady1}$ 上的良率分别作为测试集 $D'_{init2}$ 和 $D'_{steady2}$ 上的良率取值。 $D'_{init1}$ 和 $D'_{steady1}$ 上的测项良率的具体情况如表 3.11 所示。

表 3.11 仿真数据 $D'_{init1}$ 和 $D'_{steady1}$ 上各测项的良率  
Tab. 3.11 The yields of each test item on simulation data  $D'_{init1}$  and  $D'_{steady1}$

测项符号	$D'_{init1}$ 上的不良率取值/%	$D'_{steady1}$ 上的不良率取值/%
1	0.0200	0.0233
2	0.0250	0.0233
3	0.0150	0.0333
4	0.0400	0.0233
5	0.0250	0.0233
6	0.0100	0.0233
7	0.0100	0.0233
8	0.0200	0.0133
9	0.0500	0.0200
10	0.0200	0.0233
11	0.0150	0.0200
12	0.0250	0.0200
13	0.0150	0.0300
14	0.0450	0.0467
15	0.0100	0.0267
16	0.0300	0.0400

### 3.5.2 实验过程与分析

为验证包含元件互连信息的主板可靠性分析方法和传统的主板可靠性方法的准确性，分别在数据集 $D'$ 上使用行业现有的测试策略设计方法、基于传统故障树分析的主板

功能测试策略设计方法以及基于包含元件互连信息的主板可靠性分析的主板功能测试策略设计方法，得到不同测试策略设计方法对应的测试策略如表 3.12 所示。

表 3.12 3 种测试策略设计方法对应的测试策略  
Tab. 3.12 Test strategies corresponding to the 3 test strategy design methods

编号	测项策略设计方法	测试策略(初始阶段和稳定阶段)
1	行业现有的测试策略设计方法	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
2	基于传统故障树分析的主板功能测试策略设计方法	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1]
3	基于包含元件互连信息的故障树分析的主板功能测试策略设计方法	[0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1]

计算这三种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平如表 3.13 所示。

表 3.13 3 种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平  
Tab. 3.13 The average total test cost, average test time and defect level corresponding to the 3 test strategy design methods

编号	$f$ (s/board)	$f_1$ (s/board)	$f_2$
1	78.895	9.775	0.003202
2	56.669	50.909	0.000268
3	52.771	33.763	0.000883

其中，基于包含元件互连信息的主板可靠性分析的主板功能测试策略设计方法的整体经济效益最优。相比于行业现有的测试策略设计方法，该方法的缺陷水平下降了 91.63%，提升了 32.7%的经济效益；相比于基于传统故障树分析的主板功能测试策略设计方法，该方法降低了 6.9%测试时间。由于所提方法使用的是包含元件互连信息的故障树，而方法二中使用的是普通故障树，所以包含元件互连信息的故障树能够更加准确地分析主板的可靠性，更适合用于测试策略设计。

3.2.2 节所提出的自顶向下的主板可靠性分析方法没有使用限制，因此，根据该方法所获得的可靠性分析结果来设定测试策略，可以完全替代行业现有的测试策略设计方法，并且能够在一定程度上提升测试质量。而自底向上的主板可靠性分析方法只有当返修数据充足时才可使用，即可在主板测试的稳定阶段使用，所以该方法需要和自顶向下的主板可靠性分析方法或行业现有的测试策略设计方法结合使用，即为本章所提出的基于可靠性分析的主板功能测试策略设计方法。

为验证所提方法在可靠性分析过程中将两个可靠性分析方法结合使用的必要性，需要在数据集  $D'$  上做四组实验，分别对应 4 种测试策略设计方法：在  $D'_{init2}$  和  $D'_{steady2}$  均

使用行业现有的阈值设定方法、分别使用行业现有的阈值设定方法和自底向上的主板可靠性分析方法、均使用自顶向下的主板可靠性分析方法以及分别使用自顶向下的主板可靠性分析方法和自底向上的主板可靠性分析方法。

接下来分别计算各测项在行业现有阈值设定方法、自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法下的不良率阈值。行业内现有的阈值设定方法是将所有测项的不良率阈值都设定为 0.05%。后两种阈值设定方法需要基于数据或专家经验分析主板的可靠性，然后结合可靠性分析结果设定阈值。

使用自底向上的主板可靠性分析方法的前提是积累了足够量的返修数据，所以该方法只能在稳定阶段使用。首先根据初始阶段积累的返修数据，统计出每个基本事件的发生概率如表 3.14 所示。

**表 3.14 基本事件发生概率**  
**Tab. 3.14 The occurrence probability of basic events**

基本事件符号	发生概率/%	基本事件符号	发生概率/%
X <sub>1</sub>	0.0022	X <sub>17</sub>	0.0023
X <sub>2</sub>	0.0023	X <sub>18</sub>	0.0023
X <sub>3</sub>	0.0031	X <sub>19</sub>	0.0025
X <sub>4</sub>	0.0019	X <sub>20</sub>	0.0022
X <sub>5</sub>	0.0030	X <sub>21</sub>	0.0021
X <sub>6</sub>	0.0019	X <sub>22</sub>	0.0022
X <sub>7</sub>	0.0024	X <sub>23</sub>	0.0019
X <sub>8</sub>	0.0023	X <sub>24</sub>	0.0021
X <sub>9</sub>	0.0023	X <sub>25</sub>	0.0019
X <sub>10</sub>	0.0023	X <sub>26</sub>	0.0022
X <sub>11</sub>	0.0030	X <sub>27</sub>	0.0025
X <sub>12</sub>	0.0019	X <sub>28</sub>	0.0021
X <sub>13</sub>	0.0022	X <sub>29</sub>	0.0020
X <sub>14</sub>	0.0022	X <sub>30</sub>	0.0020
X <sub>15</sub>	0.0024	X <sub>31</sub>	0.0022
X <sub>16</sub>	0.0022	X <sub>32</sub>	0.0023

然后根据公式(3.2)和表 3.6 中集合的信息，计算出中间事件 M<sub>1</sub>-M<sub>16</sub> 的发生概率，即测项故障发生的概率，也即测项的不良率。在此基础上根据公式(3.5)计算出不良率阈值，假设目前对主板的质量要求较高，则  $\delta_j$  取负值，根据专家经验设定  $\delta_j = -5 \times 10^{-6}$ ，使得

对测项  $M_j$  的测试更加严格。

在自顶向下的主板可靠性分析方法中, 专家经验认为各类基本事件发生的概率成比例, 并且  $r_1=1.3$ ,  $r_2=0.9$ , 将  $r_1$  和  $r_2$  的具体取值代入公式(3.4), 得到仅包含变量  $p$  的表达式。参考目前行业内要求的笔记本主板在流向下一环节的故障概率不超过 0.05%, 设定公式(3.4)中  $T=0.05\%$ , 计算得到  $p=0.0021\%$ 。最后将  $r_1$ 、 $r_2$  和  $p$  的结果代入表 3.8 中各中间事件不良率阈值的表达式, 得到各测项的不良率阈值。

通过自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法获得各测项的不良率阈值如表 3.15 所示。

**表 3.15 自底向上和自顶向下的主板可靠性分析方法对应的不良率阈值**  
**Tab. 3.15 Failure rate thresholds corresponding to the bottom-up and the top-down motherboard reliability analysis methods**

编号	功能测项	测项不良率阈值/%	
		自底向上的主板 可靠性分析方法	自顶向下的主板 可靠性分析方法
1	无线广域网功能测项	0.0282	0.0264
2	键盘功能测项	0.0186	0.0175
3	风扇功能测项	0.0185	0.0175
4	G-sensor 功能测项	0.0186	0.0175
5	盒盖功能测项	0.0183	0.0175
6	触摸板功能测项	0.0185	0.0175
7	指点杆功能测项	0.0183	0.0175
8	HDMI 功能测项	0.0282	0.0264
9	LVDS 功能测项	0.0284	0.0264
10	指纹功能测项	0.0281	0.0264
11	摄像头功能测项	0.0280	0.0264
12	面板功能测项	0.0283	0.0264
13	蓝牙功能测项	0.0281	0.0264
14	无线局域网功能测项	0.0259	0.0245
15	扬声器功能测项	0.0301	0.0283
16	麦克风功能测项	0.0300	0.0283

上述过程分别得到行业内现有的阈值设定方法、自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法对应的不良率阈值。接下来计算仿真数据里初始阶段  $D'_{init2}$  和稳定阶段  $D'_{steady2}$  中主板在不同测试策略设计方法下的主板平均测试总成本、

主板平均测试时间和缺陷水平。某阶段的主板在某种测试策略设计方法下的主板平均测试总成本计算过程包括：首先，通过比较主板功能测试阶段中主板的实际不良率与其对应的不良率阈值得到测试策略；接下来计算使用该测试策略时，测试主板所花费的平均测试时间和假阳性主板数目；然后通过假阳性主板数目计算主板测试后的缺陷水平；最后，将测试时间和假阳性主板造成的维修时间求和，即可计算出测试总成本。以第一种测试策略设计方法为例，详细描述三个评价指标参数的计算过程。其他测试策略设计方法对应的指标参数的计算过程与第一种测试策略设计方法大体一致。

第一种测试策略设计方法是在 $D'_{init2}$ 和 $D'_{steady2}$ 上均使用行业现有的阈值设定方法。首先计算在 $D'_{init2}$ 上使用行业现有的阈值设定方法时对应的三项指标值：将各测项的不良率阈值 0.05%分别和表 3.10 中第二列 ( $D'_{init1}$ 的不良率取值) 进行比较，利用公式(3.6-3.7)得到各测项的测试策略。再根据公式(3.10)计算该测试策略下测试 $D'_{init2}$ 中主板所花费的时间，根据公式(3.11-3.13)计算假阳性主板数目和缺陷水平，然后利用公式(3.14)计算出在 $D'_{init2}$ 上使用行业现有的阈值设定方法所花费的测试总成本。接下来计算在 $D'_{steady2}$ 上使用行业现有的阈值设定方法时对应的三项指标值：将 0.05%分别和表 3.10 中第三列 ( $D'_{steady1}$ 的不良率取值) 进行比较，得到各测项的测试策略。再根据公式(3.10)计算该测试策略下测试 $D'_{steady2}$ 中主板所花费的测试时间，根据公式(3.11-3.13)计算使用该测试策略时的缺陷水平，然后利用公式(3.14)将测试时间和假阳性主板造成的维修时间求和，即可计算出测试总成本。最后将 $D'_{init2}$ 和 $D'_{steady2}$ 的三项指标值分别求和，得到仿真数据中主板使用第一种测试策略设计方法时对应的指标值。

分别在数据集 $D'$ 上使用上述 4 种测试策略设计方法，得到不同测试策略设计方法对应的测试策略如表 3.16 所示。

**表 3.16 4 种测试策略设计方法所对应的测试策略**  
**Tab. 3.16 Test strategies corresponding to the 4 test strategy design methods**

编号	测项策略设计方法	测试策略(初始阶段和稳定阶段)
1	行业现有的测试策略设计方法	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
2	行业现行的测试策略设计方法+ 自底向上的主板可靠性分析方法	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1] [0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1]
3	自顶向下的主板可靠性分析方法	[0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1]
4(本章所提方法)	自顶向下的主板可靠性分析方+ 自底向上的主板可靠性分析方法	[0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1] [0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1]

通过上述计算可以得到不同测试策略设计方法对应的平均测试总成本、平均测试时

间和缺陷水平如表 3.17 所示。

对比表 3.17 中测试策略设计方法 1 和 3 的实验结果可以看出后者漏检主板数目显著减少, 这说明方式 3 的拦截不良主板的能力高于方式 1; 并且测试总成本有所下降, 所以自顶向下的主板可靠性分析方法比行业现有的阈值设定方法更有优势。通过对比测试策略设计方法 2 和 4 的实验结果可以看出后者漏检主板数目有所减少, 并且测试总成本有所下降。由于测试策略设计方法 3 和 4 的区别仅在于在初始阶段使用了不同的阈值设定方法, 分别为行业现有的阈值设定方法和自顶向下的主板可靠性分析方法, 所以也说明自顶向下的主板可靠性分析方法比行业现有的阈值设定方法具有更强的拦截不良主板的能力。

**表 3.17 4 种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平**  
**Tab. 3.17 The average total test cost, the average test time and the defect level corresponding to the four test strategy design approaches**

编号	$f$ (s/board)	$f_1$ (s/board)	$f_2$
1	78.895	9.775	0.003020
2	53.067	30.027	0.001070
3	55.783	39.367	0.000763
4	52.771	33.763	0.000883

对比表 3.17 中测试策略设计方法 1 和 2 的实验结果可以看出后者漏检主板数目显著减少并且测试总成本更少, 这表明测试策略设计方法 2 拦截不良主板的能力高于方式 1, 两种测试策略设计方法的区别为在稳定阶段使用了不同的阈值设定方法, 分别为行业现有的阈值设定方法和自底向上的阈值设定方法。对比测试策略设计方法 3 和 4 的实验结果可以看出后者的平均测试总成本更少, 而测试策略设计方法 3 和 4 的区别与测试策略设计方法 1 和 3 的区别一致, 均为在稳定阶段使用了不同的阈值设定方法, 所以两组对比均说明在稳定阶段使用自底向上的阈值设定方法比使用行业内现有的阈值设定方法或自顶向下的阈值设定方法效果更好。

最后, 对比表 3.17 中 4 种测试策略设计方法的实验结果, 可以看出测试策略设计方法 4 对应测试总成本为 4 种方式中最小的, 这表明本章所提方法将两种可靠性分析方法结合使用的必要性, 它能够有效地提升测试质量, 并且降低平均测试总成本。

### 3.6 本章小结

本章提出了一种基于元件互连信息的主板可靠性分析方法。首先面向主板功能测试

环节分别建立了普通故障树和包含元件互连信息的故障树。然后提出了两种主板可靠性分析方法，其中，自底向上的主板可靠性分析方法能够在主板返修数据充足时对主板的可靠性进行分析；而自顶向下的主板可靠性分析方法能够在主板类型切换时导致主板返修数据不充足的场景下对主板进行可靠性分析。实验证明包含元件互连信息的故障树相较于普通故障树，能够进一步提高可靠性分析的准确性。后者解决了目前研究中维修数据不足的情况下，难以对主板进行准确的可靠性分析的难题。

主板的可靠性分析结果能够用于主板功能测试阶段的测试策略设计。利用主板可靠性分析结果为主板的测项设定良率阈值，然后通过控制测项的测试状态将主板测项的良率控制在可靠性范围之内，从而保证主板的测试质量。通过对比实验和消融实验，验证了所提方法能够为不同阶段设定更加合理的良率阈值，更加灵活地调整了测试策略，该测试策略设计方法在保证测试质量的同时有效的降低了测试总成本。

## 第四章 融合决策树和可靠性分析的可解释测试策略设计方法

行业现行的主板功能测试策略设计方法可解释性差，并且未考虑机理信息，所以往往存在过拟合问题。为解决这一问题，本文提出了一种融合决策树和可靠性分析的可解释测试策略设计方法。首先利用主板可靠性分析方法获得主板中各测项的可靠性指标范围。然后在该指标范围内生成第一层决策树模型的训练数据，根据训练出的决策树模型缩小各测项的可靠性指标范围并进行特征选择。接下来，根据第一层决策树模型的训练结果准备第二层决策树的训练数据，并根据训练结果为主板设计测试策略。实验结果表明，所提方法具有较强可解释性，并且能够充分利用测试结果中的信息，显著提升了整体经济效益。

### 4.1 方法介绍

#### 4.1.1 整体流程

第 3.3 节将可靠性分析结果用于测试策略设计，能够保证主板的可靠性，并且在一定程度上提升了主板的测试效率。但是，上述过程不能充分利用主板测试数据所包含的信息，导致难以达到最佳的经济效益。此外，行业现有的测试策略设计方法未融入机理信息，往往存在过拟合问题，并且可解释性较差。为解决上述问题，本章提出一种融合决策树和可靠性分析的可解释测试策略设计方法。该方法能够充分利用主板测试结果，使得测试效率显著提升。此外，由于该方法是基于决策树模型的，其二叉结构正好对应测试策略的两种取值，这使得该方法所设计出的测试策略具有强可解释性；由于该方法将机理信息和数据驱动模型相融合，这使得该方法能够缓解纯数据驱动模型的过拟合问题。该方法的具体流程如下：

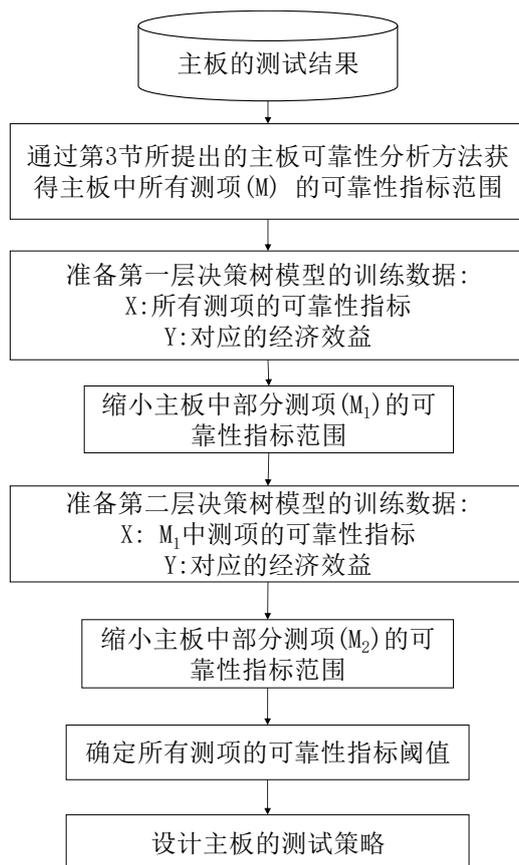
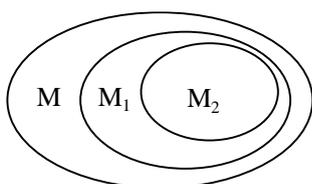


图 4.1 融合决策树和可靠性分析的可解释测试策略设计方法流程图

Fig. 4.1 The flowchart of the interpretable test strategy design method based on decision tree and reliability analysis

图 4.1 中, 集合  $M$  包含了主板的所有测项, 集合  $M_1$  是所训练出的第一层决策树模型中目标路径上出现的所有测项的集合, 集合  $M_2$  是所训练出的第二层决策树模型中目标路径上出现的所有测项的集合。因此, 主板测项所组成的集合  $M$ 、 $M_1$  和  $M_2$  的关系如图 4.2 所示。



- M: 主板中所有的测项组成的集合 (测项个数是 $m$ )
- $M_1$ : 第一层决策树中目标路径上出现的测项组成的集合 (测项个数是 $m_1$ )
- $M_2$ : 第二层决策树中目标路径上出现的测项组成的集合 (测项个数是 $m_2$ )

图 4.2 主板测项所组成的集合  $M$ 、 $M_1$  和  $M_2$  的关系

Fig. 4.2 The relationship between the sets  $M$ ,  $M_1$  and  $M_2$  formed by the test items

本章所提方法在纯决策树模型的基础上不断改进。首先将纯决策树模型用于测试策略设计。然后为了缓解该模型的过拟合问题, 在该模型中加入机理信息, 提出了融合决策树和可靠性分析发主板功能测试策略设计方法。最后为了进一步提升主板功能测试环节的整体经济效益, 再加入一层决策树模型, 提出了融合双层决策树和可靠性分析发主

板功能测试策略设计方法。

#### 4.1.2 基于决策树的主板功能测试策略设计方法

为解决行业线性的测试策略设计方法缺乏理论依据的问题，本文提出了基于决策树的主板功能测试策略设计方法，该方法基于数据驱动模型对各测项的良率阈值进行了合理的设定。它通过决策树模型将大量的良率阈值组合进行分类，找到可以达到最高经济效益的阈值组合的特征，根据该特征对主板功能测项的阈值进行设定，从而得到更加高效的测试策略。

基于决策树的主板功能测试策略设计方法的流程如图 4.3 所示，其步骤包含四个阶段：首先划分历史数据，然后准备决策树模型的训练数据，接下来设定具体的训练参数进而训练决策树，最后确定测试策略。

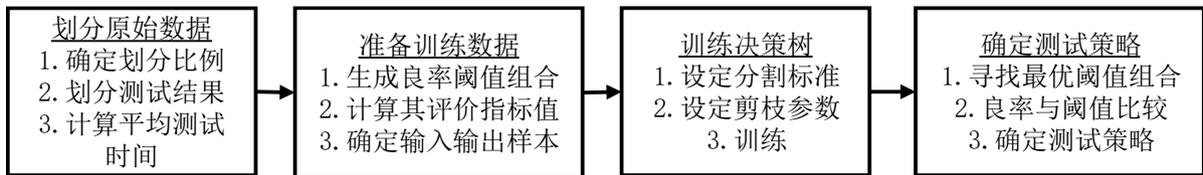


图 4.3 基于决策树的阈值设定方法的流程图

Fig. 4.3 The flowchart of the decision tree based threshold setting approach

首先确定一个划分比例将历史数据中主板的测试结果划分为两部分，第一部分数据需要做进一步处理，作为决策树模型的训练数据，第二部分数据作为确定决策树目标路径的验证数据。

由于决策树模型的目的在于为主板的测项设定合适的良率阈值，使得主板功能测试阶段的整体经济效益较高或者测试时间较短，因此决策树模型的输入数据是不同测项的良率阈值所组成的向量，记为  $\mathbf{x} = [x_1, x_2, \dots, x_k]^T$ ，输出数据是不同的良率阈值取值所对应的经济效益，记为  $\mathbf{y} = [f_1, f_2, \dots, f_k]^T$ ，其中  $k$  是主板中功能测项的个数。

由于良率阈值在良率的范围内波动，因此首先需要根据测项的测试结果统计出对应的良率取值范围，作为良率阈值取值范围。为包含良率阈值的所有取值情况，某测项的良率阈值取值在良率阈值范围内随机生成，主板中各测项的良率阈值构成一个良率阈值组合，是决策树模型的一个输入数据。由于经济效益是连续值，为避免过拟合，需要按照整体经济效益或者测试时间对样本进行聚类，最终将主板功能测项的良率阈值组合和对应的经济效益聚类标签作为决策树模型的训练样本。

在步骤三中为决策树的训练选择合适的分割标准和剪枝参数。分割标准用来选择划

分特征，使得最大限度地将数据分类；剪枝参数能有效的防止过拟合现象，使得分类的结果更可靠。

在步骤四中，根据验证数据在训练好的决策树模型中寻找目标路径，该目标路径上的良率阈值组合使得主板功能测试阶段的整体经济效益最高或者测试时间最短。最后将良率阈值和良率比较，得到对应的测试策略。

基于决策树模型为主板功能测试阶段设计测试策略的具体过程如下：

第 3.4.1 节中介绍到，某典型笔记本主板的一批测试结果为  $\mathbf{D}$ ，每块主板的各个测试项目的测试时间为  $\mathbf{T}$ 。

首先将测试结果按照划分比例  $\mathbf{D}$  划分为  $\mathbf{D}_{train}$  和  $\mathbf{D}_{valid}$ 。 $\mathbf{D}_{train}$  用来准备决策树的训练样本， $\mathbf{D}_{valid}$  作为验证集用来确定决策树模型中的目标路径。

接下来根据历史数据  $\mathbf{D}_{train}$  确定各测试项目的良率阈值范围，然后在良率阈值范围内生成多组阈值组合。具体地，将  $\mathbf{D}_{train}$  分成  $s$  个矩阵，如下所示：

$$\mathbf{D}_{train} = (\mathbf{D}_{train}^1 \quad \mathbf{D}_{train}^2 \quad \cdots \quad \mathbf{D}_{train}^s),$$

其中  $\mathbf{D}_{train}^i$  表示  $\mathbf{D}$  被划分后的第  $i$  个矩阵，同样  $\mathbf{D}_{train}^i \in \mathbb{Z}_2^n \times m$ ， $\mathbb{Z}_2 = \{0,1\}$ 。然后分别计算每个矩阵  $\mathbf{D}_{train}^i (i = 1, 2, \dots, s)$  中各测项的良率，计算过程如下所示：

$$y_j = \frac{\sum_{k=1}^n (\mathbf{D}_{train,j,k}^i = 1)}{n}, i = 1, \dots, s, j = 1, \dots, m, \quad (4.1)$$

其中， $\mathbf{D}_{train,j,k}^i$  表示矩阵  $\mathbf{D}_{train}$  分割后的第  $i$  个矩阵里第  $k$  块主板的第  $j$  个测试项目的测试结果。

利用公式(4.1)计算得到  $s$  个矩阵中良率的最大值和最小值，分别记为  $y_{max}$  和  $y_{min}$ ，所以测项的良率范围是  $[y_{min}, y_{max}]$ ，将该区间作为该测项的良率阈值设定范围。

在每个测项的良率阈值设定范围内随机生成  $k$  个良率阈值，并组合成矩阵，记为  $\mathbf{X} = \mathbf{R}^{k \times m}$ ，将  $\mathbf{X}$  作为决策树模型的输入数据。其中，矩阵中的一行由  $m$  个测项的良率阈值组成，称为一个良率阈值组合，第  $i$  组阈值组合表示为  $\mathbf{x} = [x_{i,1}, x_{i,1}, \dots, x_{i,m}]$ 。对于每组阈值，依据公式(3.6-3.7)确定主板的测试策略。

由于阈值组合对应的测试成本、返修时间等是连续变化的量，我们首先需要通过聚类方法将测试成本或返修时间进行聚类，进而使每个阈值组合对应一个聚类标签，将阈

值和对应的聚类标签分别作为决策树的输入输出样本。

根据工厂的实际情况，制定测试策略的目标不同。当允许的生产环节至测试环节的最长时间间隔较短，即生产进度较快时，需要加快功能测试产线上的测试速度，此时测试成本是关键的经济效益指标，因此将测试成本作为特征进行聚类，选择测试成本较低的类别对应的测试策略；当对漏检的不良主板数量有严格要求并且测试时间充足时，此时关键的经济效益指标是测试成本和返修时间，因此将测试成本和返修时间同时作为特征一起聚类，选择测试成本满足要求而返修时间最短的类别对应的测试策略。其中，测试成本所需要满足的要求是：该测试成本不高于行业现有的测试策略对应的测试成本。

所以，该方法采用两种聚类方式：将上述的测试成本作为特征进行聚类；将测试成本和返修时间同时作为特征一起聚类。前者考虑了测试成本，本质上考虑了测试时间和返修时间的整体性能；后者可以将返修时间单独考虑，在保证整体效益满足要求的基础上大幅提升测试质量，故后者更适合对漏检主板数有严格要求的情况。

利用公式(3.10-3.14)分别计算第  $k$  组测试策略所对应的指标参数，将平均测试总成本记为  $\mathbf{y} = [f(1), f(2), \dots, f(k)]^T$ ，将平均测试时间记为  $\mathbf{y}_1 = [f_1(1), f_1(2), \dots, f_1(k)]^T$ 。

使用 K-means 方法将  $\mathbf{y}$  或  $\mathbf{y}_1$  聚成  $l$  类，聚类标签由  $[0, 1, 2, \dots, l]$  表示， $\mathbf{y}$  对应的标签记为  $\mathbf{l} = [l(1), l(2), \dots, l(k)]^T$ ， $\mathbf{y}_1$  对应的标签记为  $\mathbf{l}_1 = [l_1(1), l_1(2), \dots, l_1(k)]^T$ ，作为决策树模型的输出数据。

当聚类方式是以平均总测试成本为聚类特征时，决策树模型的训练数据  $\mathcal{A}$  如下所示：

$$\mathcal{A} = [\mathbf{X} \mid \mathbf{l}] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} & l(1) \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} & l(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{k,1} & x_{k,2} & \cdots & x_{k,m} & l(k) \end{bmatrix} \quad (4.2)$$

当聚类方式为以测试成本作为特征进行聚类时，决策树模型的训练数据  $\mathcal{A}$  如下所示：

$$\mathcal{A} = [\mathbf{X} \mid \mathbf{l}] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} & l_1(1) \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} & l_1(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{k,1} & x_{k,2} & \cdots & x_{k,m} & l_1(k) \end{bmatrix} \quad (4.3)$$

在训练决策树之前，需要为决策树模型设定分割标准和剪枝策略。分割标准可以选择信息熵或者基尼指数方式设定；剪枝参数可以按照 2.2 节中介绍的先剪枝或者后剪枝

方式设定。

最后，在决策树中寻找目标叶节点所对应的路径的阈值组合。不同测试需求场景下，选用不同的聚类方式，训练出的决策树不同，其目标叶节点也不同。将测试成本作为特征进行聚类时，假设聚类成“低”、“中”和“高”三类，目标叶节点则是测试成本对应最小的类别标签，即“低”标签；将测试成本和返修时间同时作为特征进行聚类时，由于该聚类方式更加注重测试质量，所以目标叶节点应该是返修时间最短、测试成本最短或中等所对应的类别标签组合。

首先在决策树中根据目标叶节点筛选出符合要求的路径，路径上的非叶节点的内容如公式(2.5)所示，所以不同路径会对应不同的阈值范围组合。先利用公式(4.4)将路径上的非叶节点所表示的阈值范围转换成某一确定的阈值，该阈值作为对应测项的良率阈值，其中， $j$ 指出现在路径中的测项的序号，所以 $T_{j,lb}$ 和 $T_{j,ub}$ 分别表示第 $j$ 个测试项目的阈值上下限。重复该步骤即可确定路径上所有非叶节点所对应的测项的良率阈值，所有测项的阈值根据测项的阈值范围确定后，即可得到一个阈值组合。

$$T_j = \begin{cases} \frac{c + T_{j,lb}}{2}, & \text{路径中该结点的子结点是左节点} \\ \frac{c + T_{j,ub}}{2}, & \text{路径中该结点的子结点是右节点} \end{cases} \quad (4.4)$$

再利用公式(3.7)将阈值组合和良率组合比较，得到所有测项的测试状态。这些测试状态即为该路径对应的测试策略。

在获取阈值组合的过程中可能会遇到测项没有在路径上的全部出现的问题，已出现的测项可以根据公式(4.5)确定其阈值，对于未出现的测项可以根据专家经验设定其阈值，若对测试质量有严格要求，则目标是测试该项；若对测试质量没有严格要求，则目标是不测试该项。所以对于未出现的测项的阈值的设定方法如下：

$$T_j = \begin{cases} T_{j,ub}, & \text{计划测试该项} \\ T_{j,lb}, & \text{计划不测该项} \end{cases} \quad (4.5)$$

其中， $j$ 指未出现在路径中的测项的序号， $T_j$ 表示测项 $M_j$ 的良率阈值。

当有多条路径同时符合满足时，我们需要从诸多路径中挑选出最优的路径。在训练过程中经过剪枝操作，避免了过拟合，所以，最终满足要求的路径都具有参考价值。

我们需要在验证集 $D_{valid}$ 上验证所有符合要求的路径对应的测试策略，最后挑选出

最优的测试策略。两种聚类方式下，最优测试策略的选择方式不同。

当以测试成本为特征聚类时，首先将通向目标叶节点的路径对应的阈值组合找出，并获得对应的测试策略，然后在矩阵 $D_{valid}$ 上计算该测试策略下的测试成本，选择测试成本最低对应的测试策略，即为最优的测试策略；当以测试成本和返修时间为特征同时聚类时，同样的在矩阵 $D_{valid}$ 上计算该测试策略下的测试成本和漏检主板数目，选择测试成本满足要求并且漏检主板数目最小所对应的测试策略，即为最优的测试策略。

#### 4.1.3 融合决策树和可靠性分析的主板功能测试策略设计方法

在 4.1.2 节中所提方法仅利用统计学知识在历史数据上直接计算良率，该方法严重依赖于历史数据，但是在主板测试初期，历史数据缺乏，无法准确统计出良率的范围；此外，由于测项发生故障具有一定的偶然性，这导致历史数据不稳定，所以根据统计的方法不能准确计算出实际良率。

为此，我们提出一种融合故障树和决策树的主板功能测试策略设计方法，利用第 3 章所提出的主板可靠性分析方法获得各测项的良率阈值设定范围的方法，该方法基于对 SMT 生产过程和主板结构的机理分析，严格分析造成各测试项目故障的原因，建立了专门针对主板功能测试环节的故障树。再根据专家经验确定中间事件发生的概率范围，计算出各测试项目的良率范围。相比于 4.1.2 节中的方法，该方法能够减少历史数据不稳定所造成的影响，更适合用于计算主板中各测试项目的良率范围。

在主板维修数据缺乏和充足的情况下，分别使用自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法，确定主板的良率阈值设定范围。然后再按照 4.1.2 节中描述的步骤，训练决策树模型，为主板各测项设定高效的良率阈值。最终利用公式(3.6-3.7)确定主板的测试策略。

#### 4.1.4 融合双层决策树和可靠性分析的主板功能测试策略设计方法

本节提出一种基于双层决策树和可靠性分析的测试策略设计方法，由于该方法在第 4.1.3 节中所提方法的基础上加了一层决策树结构，第二层决策树能够在第一层决策树模型的基础上对整体经济效率或者测试效率优中选优，并且能够根据最新历史数据对测试策略做进一步调整，所以该方法能够在融合决策树和故障树分析的主板功能测试策略设计方法的基础上进一步提升整体经济效益。

基于双层决策树的主板功能测试策略设计方法包括以下步骤：首先利用主板维修数

据和主板的故障树结构对主板进行可靠性分析，初步确定良率阈值设定范围；再根据主板的可靠性分析结果生成第一层决策树模型的训练数据，并设定模型的训练参数，训练第一层决策树，获得经济效益最优对应的阈值设定范围，并进行特征选择；然后根据第一层决策树模型所获得的新阈值设定范围生成第二层决策树模型的训练数据，训练第二层决策树，进一步缩小良率阈值设定范围；最后为各测项设定合适的良率阈值，将实际产线上主板的良率与上述良率阈值作比较，得到主板功能测试策略。基于双层决策树和可靠性分析的可解释测试策略设计方法的具体的流程如下：

首先，根据第 3 章提出的主板可靠性分析方法，得到主板中各测项的良率范围  $[\beta_{min}(M_j), \beta_{max}(M_j)]$ 。再根据专家经验设定  $\delta(M_j)$ ，利用公式(3.5)确定主板中各测项的良率阈值设定范围  $[t_{min}(M_j), t_{max}(M_j)]$ 。

接下来，按照一定的划分比例将主板的历史测试数据划分为三部分，分别记为  $D_{train1}$ 、 $D_{train2}$  和  $D_{valid}$ ，然后在数据集  $D_{train1}$  上按照第 5.1.1 节所介绍的过程准备第一层决策树的训练数据，并训练第一层决策树，利用  $D_{valid}$  中的数据确定目标路径。

第一层决策树模型的目标路径中出现  $m_1$  个测项，这些测项组成集合  $M_1$ 。目标路径中一些测项未出现，意味着这些测项良率阈值的设定不会影响主板功能测试环节的整体经济效益或者测试效率，所以这些测项不会进入第二层决策树的训练，该过程对主板特征进行了特征选择。根据公式(4.4)对这  $m_1$  个测项的良率阈值设定范围进行计算，记中的第  $j$  个测项的良率阈值设定范围为  $[\tilde{\beta}_{min}(M_{1,j}), \tilde{\beta}_{max}(M_{1,j})]$ ，其中  $M_{1,j} \in M_1$ 。

在良率阈值设定范围  $[\tilde{\beta}_{min}(M_{1,j}), \tilde{\beta}_{max}(M_{1,j})]$  内随机生成  $k$  个良率阈值，并组合成矩阵，记为  $X_1 = R^{k \times m_1}$ ，将  $X_1$  作为第二层决策树模型的输入数据。首先确定每组良率阈值组合所对应的经济效益，然后将样本聚成 3 类，从而确定第二层决策树模型的训练数据，每个样本对应的聚类标签是  $l' = [l'(1), l'(2), \dots, l'(k)]^T$ ， $y_1$  对应的标签记为  $l_1 = [l'(1), l'(2), \dots, l'(k)]^T$ ，其形式如下：

当聚类方式是以平均总测试成本为聚类特征时，决策树模型的训练数据  $\mathcal{A}$  如下所示：

$$\mathcal{A} = [\mathbf{X} | \mathbf{1}] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m_1} & l'(1) \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m_1} & l'(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{k,1} & x_{k,2} & \cdots & x_{k,m_1} & l'(k) \end{bmatrix} \quad (4.6)$$

当聚类方式为以测试成本作为特征进行聚类时，决策树模型的训练数据  $\mathcal{A}$  如下所示：

$$\mathcal{A} = [\mathbf{X} | \mathbf{1}] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m_1} & l'_1(1) \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m_1} & l'_1(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{k,1} & x_{k,2} & \cdots & x_{k,m_1} & l'_1(k) \end{bmatrix} \quad (4.7)$$

训练第二层决策树并根据决策树模型对经济效益优中取优，进一步缩小测项的良率阈值设定范围。第二层决策树模型中有  $m - m_1 - m_2$  个测项未出现，剩下的  $m_2$  个测项组成测项集合  $\mathbf{M}_2$ 。经济效益最优所对应的  $m_2$  个测项良率阈值设定范围记为  $[\tilde{\beta}_{\min}(M_{2,j}), \tilde{\beta}_{\max}(M_{2,j})]$ ，其中  $M_{2,j} \in \mathbf{M}_2$ 。

综合双层决策树的训练结果，根据良率阈值设定范围设定准确的良率阈值  $\beta(M_j)$ ，如公式所示：

$$\beta(M_j) = \begin{cases} \frac{\tilde{\beta}_{\min}(M_{2,j}) + \tilde{\beta}_{\max}(M_{2,j})}{2}, & M_{2,j} \in \mathbb{M}_2 \\ \frac{\tilde{\beta}_{\min}(M_{1,j}) + \tilde{\beta}_{\max}(M_{1,j})}{2}, & M_{1,j} \in \mathbb{M}_1 \text{ and } M_{1,j} \notin \mathbb{M}_2 \\ \frac{\beta_{\min}(M_j) + \beta_{\max}(M_j)}{2}, & M_j \in \mathbb{M} \text{ and } M_j \notin \mathbb{M}_1 \end{cases} \quad (4.8)$$

其中， $M_{2,j} \in \mathbf{M}_2$  表示在第二层决策树模型中出现的测项， $M_{1,j} \in \mathbf{M}_1$  and  $M_{1,j} \notin \mathbf{M}_2$  表示在第一层决策树模型中出现但在第二层决策树模型中未出现的测项， $M_j \in \mathbf{M}$  and  $M_j \notin \mathbf{M}_1$  表示在第一层决策树模型中未出现的测项。

最后，利用公式(3.6-3.7)将主板测试过程中测项的实际良率与良率阈值作比较得到主板测试策略  $s$ 。

## 4.2 实验与分析

### 4.2.1 数据准备

首先利用公式(3.9)计算实际数据中各测项所需要的平均测试时间，将其作为仿真数

据中主板各测试项目的测试时间，记为 $\mathbf{t} \in \mathbf{R}^m$ 。其中， $\mathbf{t}$ 是长度为 $m$ 的一维向量， $m$ 是主板的测项数目， $\mathbf{R}$ 代表实数。

仿真数据准备好后，需要将其划分成三部分。首先确定仿真数据的划分比例为6:2:2，将测试结果 $\mathbf{D}$ 按划分比例划分为 $\mathbf{D}_{train}$ 、 $\mathbf{D}_{valid}$ 和 $\mathbf{D}_{test}$ ，分别用作决策树的训练、方案的选择和方案效果测试。

根据专家经验，测项的良率阈值随着其良率波动，所以接下来将各测项的良率作为参考，设定该测项的良率阈值的波动范围。分别用4.1.1节所介绍的统计方法和4.1.2节所介绍的机理方法获得良率范围，从而确定良率阈值的波动范围。首先根据专家经验确定各底事件发生的概率范围如下：

**表 4.1 底事件的发生概率范围**

**Tab. 4.1 Probability ranges for the occurrence of bottom events**

底事件符号	发生概率/DPPM	底事件符号	发生概率/DPPM
X <sub>1</sub>	[29.30,121.70]	X <sub>17</sub>	[22.75,99.75]
X <sub>2</sub>	[23.20,100.20]	X <sub>18</sub>	[29.51,121.91]
X <sub>3</sub>	[21.87,91.17]	X <sub>19</sub>	[24.57,101.57]
X <sub>4</sub>	[25.88,118.28]	X <sub>20</sub>	[28.80,121.20]
X <sub>5</sub>	[20.82,90.12]	X <sub>21</sub>	[27.45,119.85]
X <sub>6</sub>	[25.95,118.35]	X <sub>22</sub>	[28.72,121.12]
X <sub>7</sub>	[23.86,100.86]	X <sub>23</sub>	[26.07,118.47]
X <sub>8</sub>	[23.41,100.41]	X <sub>24</sub>	[28.16,120.56]
X <sub>9</sub>	[23.18,100.18]	X <sub>25</sub>	[26.17,118.57]
X <sub>10</sub>	[22.69,99.69]	X <sub>26</sub>	[29.38,121.78]
X <sub>11</sub>	[20.97,90.27]	X <sub>27</sub>	[24.85,101.85]
X <sub>12</sub>	[25.68,118.08]	X <sub>28</sub>	[27.69,120.09]
X <sub>13</sub>	[28.93,121.33]	X <sub>29</sub>	[27.26,119.66]
X <sub>14</sub>	[28.71,121.11]	X <sub>30</sub>	[26.66,119.06]
X <sub>15</sub>	[23.58,100.58]	X <sub>31</sub>	[28.70,121.10]
X <sub>16</sub>	[29.08,121.48]	X <sub>32</sub>	[22.53,99.53]

其中，底事件的发生概率以 DPPM (defect part per million) 为单位，指每百万主板中底事件发生的次数。

4.1.1 节所提方法使用统计方法在数据 $\mathbf{D}_{train}$ 上计算各动态测试项目的良率的范围。先将 $\mathbf{D}_{train}$ 分成10组，再利用公式(4.1)分别计算每组中各测项的良率，最后将良率的最

大值和最小值分别作为该测项的良率阈值的上下限。4.1.2 节和 4.1.3 节所提方法均使用机理方法获得初始的良率阈值设定范围，如表 4.2 所示。

表 4.2 分别通过统计方法和机理方法获得的初始良率阈值设定范围  
Tab. 4.2 Initial yield threshold setting range obtained by statistical and mechanistic methods

编号	测项名称	实际良率	初始良率阈值设定范围	
			统计方法	机理方法
1	无线广域网功能测项	0.99968	[0.99920,1.00000]	[0.99971,0.99875]
2	键盘功能测项	0.99968	[0.99920,1.00000]	[0.99981,0.99916]
3	风扇功能测项	0.99912	[0.99860,0.99960]	[0.99981,0.99916]
4	G-sensor 功能测项	0.99938	[0.99860,1.00000]	[0.99981,0.99916]
5	盒盖功能测项	0.99940	[0.99900,0.99980]	[0.99981,0.99916]
6	触摸板功能测项	0.99972	[0.99920,1.00000]	[0.99981,0.99916]
7	指点杆功能测项	0.99968	[0.99940,1.00000]	[0.99981,0.99916]
8	HDMI 功能测项	0.99972	[0.99920,1.00000]	[0.99971,0.99875]
9	LVDS 功能测项	0.99970	[0.99960,1.00000]	[0.99971,0.99875]
10	指纹功能测项	0.99982	[0.99940,1.00000]	[0.99971,0.99875]
11	摄像头功能测项	0.99940	[0.99860,0.99980]	[0.99972,0.99875]
12	面板功能测项	0.99960	[0.99920,1.00000]	[0.99971,0.99875]
13	蓝牙功能测项	0.99916	[0.99880,0.99980]	[0.99971,0.99875]
14	无线局域网功能测项	0.99958	[0.99920,1.00000]	[0.99974,0.99887]
15	扬声器功能测项	0.99906	[0.99840,0.99940]	[0.99969,0.99863]
16	麦克风功能测项	0.99966	[0.99900,1.00000]	[0.99969,0.99863]

其中，利用 4.1.1 节所提方法计算的部分测项的良率阈值的设定上限是 100%，这表示一段时间内该测项未出现测试不通过的情况。

接下来，分别在使用统计方法和机理方法所获得的初始良率阈值设定范围内产生 50000 个随机数，组合成两个 50000 行 16 列的矩阵，分别用  $B_1$  和  $B_2$  表示。其中， $B_1$  是 4.1.1 节所提方法使用的初始阈值设定组合， $B_2$  是 4.1.2 和 4.1.3 节所提方法使用的初始阈值设定组合。

然后计算各阈值组合在  $D_{train}$  上的测试成本和返修成本。每组阈值与表 4.2 中的实际良率作比较，根据公式(3.6-3.7)得到对应的测试策略。在不同策略下，利用公式(3.10)计算出平均测试时间，根据公式(3.13)计算出缺陷水平。将测试时间和返修时间之和作为主板的测试成本，根据专家经验设定  $w=120*60s$ ，即可根据公式(3.14)计算出这 50000 块主板的评价测试总成本。

4.1.1 节描述了样本的两种聚类方式，下面以第一种聚类方式为例进行实验，即将测试策略对应的整体经济效益为聚类特征，以追求最大的经济效益，因此需要选择经济效益最高的类别作为目标路径。分别对阈值组合  $B_1$  和  $B_2$  使用 k-means 算法进行聚类。以测试整体经济效益为聚类特征，则样本的聚类结果如表 4.3 所示。

根据聚类中心的大小关系确定聚类标签的含义，如表 4.3 所示，聚类标签“中”、“低”和“高”分别对应整体经济效益为中、低和高的情况。其中，加粗的聚类标签，即为决策树模型中目标叶节点所属的类别。

表 4.3 4.1.1 节、4.1.2 节和 4.1.3 节所提方法的样本聚类结果

Tab. 4.3 Sample clustering results for the methods proposed in subsections 4.1.1, 4.1.2 and 4.1.3

编号	测试策略设计方法	聚类标签	聚类中心(s/board)	样本数
1	4.1.1 节所提方法	中	78.94	9621
		<b>高</b>	100.51	3323
		<b>低</b>	58.23	7056
2	4.1.2 节所提方法	中	84.80	6516
		<b>低</b>	53.23	7961
		高	98.66	5523
3	4.1.3 节所提方法 (本章所提方法)	高	73.62	10288
		<b>低</b>	52.42	8929
		中	61.21	783

#### 4.2.2 参数设置

将阈值组合和其对应的类别标签构成样本数据，先按照 6:2:2 的比例将数据划分为训练集、测试集和验证集，再调用 scikit-learn 机器学习工具包中的决策树方法进行训练，训练前需要确定 class\_weight、ccp\_alpha 等超参数的取值。

为解决表 4.3 中聚类样本不均衡的问题，使用 scikit-learn 机器学习工具包所提供的支持类加权的决策树算法，将 DecisionTreeClassifier 函数的 class\_weight 参数设置为“balanced”，即可自动调整不均衡样本的权重。为了防止决策树的训练结果过拟合，使

用后剪枝操作结果试验方法，设定  $ccp\_alpha = 0.33$ ，以此控制决策树的深度。

接着，将良率阈值组合和对应的标签分别作为决策树模型训练的输入和输出数据，开始训练决策树。决策树训练完成后，首先根据目标叶节点确定目标路径，然后分别根据公式(4.8)确定路径中出现的测试项目的阈值，根据公式(4.5)确定路径中未出现的测试项目的良率阈值，即可确定该路径对应的良率阈值组合。最后根据 4.1.1 节所描述的测试策略设定过程，根据良率阈值组合为主板功能测试阶段设计测试策略。

### 4.2.3 实验过程和结果分析

分别使用行业现行的测试策略设计方法、4.1.1 节所提出的基于决策树的主板功能测试策略设计方法、4.1.2 节所提出的融合决策树和可靠性分析的测试策略设计方法以及 4.1.3 节所提出融合双层决策树和可靠性分析的测试策略设计方法为主板功能测试阶段设计测试策略，测试策略如表 4.4 所示。

表 4.4 4 种测试策略设计方法对应的测试策略

Tab. 4.4 Test strategies corresponding to the 4 test strategy design methods

编号	测试策略设计方法	测试策略
1	行业现行的测试策略设计	[0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0]
2	5.1.1 节所提方法	[0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0]
3	5.1.2 节所提方法	[0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0]
4(本章所提方法)	5.1.3 节所提方法	[0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0]

在测试集  $D_{test}$  上，利用公式(2.3-3.14)分别计算表 4.4 中的 4 个测试策略所对应的三个指标值，如表 4.5 所示。

表 4.5 4 种测试策略设计方法对应的平均测试总成本、平均测试时间和缺陷水平

Tab. 4.5 The average total test cost, average test time and defect level corresponding to the 4 test strategy design methods

编号	$f(s/board)$	$f_1(s/board)$	$f_2$
1	74.056	65.194	0.003076
2	56.687	48.933	0.001076
3	54.182	40.889	0.001846
4(本章所提方法)	51.944	41.421	0.001461

比较同一种阈值设定方法下采用不同聚类方式所获得的测试策略，可见它们的优化的重点不同。两种聚类方式的聚类特征分别是测试成本、测试成本和返修时间。第一类聚类方式获得的测试策略对应的漏检主板数比第二类多，同时功能测试所需要的时间会比第二类少。所以，以测试成本为特征聚类时，旨在整体上优化测试成本；以测试成本

和返修时间为特征同时聚类时，旨在确保整体测试成本满足要求的情况下，尽可能减少不良主板的漏检，从而减少成品检测阶段的维修成本，这个过程中可能会增加测试时间。

比较不同的测试策略设计方法，其中融合双层决策树和可靠性分析方法的主板功能测试策略设计方法的整体经济效益最优。行业现行的测试策略设计方法将各测项的良率阈值始终设定为 99.95%；4.1.1 节所提出的方法基于纯数据驱动模型，未准确的分析出主板各测项的可靠性指标，因此设计的测试策略往往存在过拟合的问题；4.1.2 节所提出的方法是融合决策树和可靠性分析方法的主板功能测试策略设计方法，该方法获得了各个测项的真实良率，实际测试过程中测试结果所反映出的良率会不断地波动，而该方法所设计的测试策略也会随着实际情况做出调整，将测项和主板的良率控制在可靠性范围之内；4.1.3 节所提出的方法，也即本章所提出的方法能够在 5.1.2 节所提方法的基础上，对整体经济效益优中选优，因此取得了最佳的经济效益。

### 4.3 本章小结

本章提出了一种融合决策树和可靠性分析的可解释测试策略设计方法，将数据驱动模型和机理信息相融合。首先利用主板可靠性分析方法获得主板中各测项的可靠性指标范围，然后在该指标范围内生成不同的指标阈值组合，作为双层决策树模型训练的输入数据，接着计算指标阈值组合所对应的经济效益，聚类后作为双层决策树模型训练的输出数据。最后根据双层决策树模型确定指标阈值并设计测试策略。

本章所提出的方法所训练出的二叉决策树模型的路径正对应各功能测项的两种测试策略，因此，相比于行业现行的测试策略设计方法，所提方法的可解释性更强。对比实验表明所提方法充分利用了主板测试结果数据，能够达到更高的经济效率。此外，所提方法中两种聚类方式的侧重点不同，得到的测试策略对行业都有重要参考价值；根据所提方法训练出的决策树模型还可以对测项的测试顺序进行排序。

## 总结与展望

### 工作总结

本文的主要目的是为主板功能测试阶段设计测试策略，从而提升该阶段的整体经济效益。首先基于故障树模型对主板的可靠性进行分析，然后基于可靠性分析结果，为主板功能测试阶段设计合理的测试策略。本文主要工作和成果总结如下：

(1) 包含元件互连信息的主板可靠性分析方法。该项工作以某典型笔记本主板为例，首先针对主板功能测试环节，分别为该典型主板建立了普通故障树和包含元件互连信息的故障树。然后分别设计了自底向上的主板可靠性分析方法和自顶向下的主板可靠性分析方法。其中，自底向上的主板可靠性分析方法能够在主板维修数据充足的情况下，利用故障树分析方法定量计算各测项的可靠性指标；自顶向下的主板可靠性分析方法在主板维修数据缺乏的情况下，能够结合专家经验反推各测项的可靠性指标。最后，将基于普通故障树的主板可靠性分析结果和基于元件互连信息的主板可靠性分析结果分别用于测试策略设计，并与行业内现行的测试策略设计方法进行对比。实验结果表明包含元件互连信息的故障树能够进一步提高主板可靠性分析的准确，并且使用该可靠性分析方法所设计的测试策略能够在保证主板质量的同时，降低主板测试时间成本。

(2) 融合决策树和可靠性分析的可解释测试策略设计方法。现有的主板功能测试策略设计方法可解释性差，并且未考虑机理信息，所以往往存在过拟合问题。针对该问题，本文提出了一种融合决策树和可靠性分析的可解释测试策略设计方法。首先利用主板可靠性分析方法得到某典型笔记本主板中各测项的可靠性指标范围。然后在该指标范围内生成第一层决策树模型的训练数据，并根据训练出的决策树模型缩小各测项的可靠性指标范围，同时进行特征选择。接下来，根据第一层决策树模型的训练结果准备第二层决策树的训练数据，并根据第二层决策树模型的训练结果设计主板的测试策略。最后将所提方法和行业现行的测试策略设计方法比较，验证了所提方法能够充分利用机理信息，缓解过拟合问题，从而显著提升整体经济效益，同时该方法根据训练出的决策树模型设计测试策略，具有较强的可解释性。

### 本文创新点

本文的研究内容为主板可靠性分析提供了理论基础，并且通过设计合适的测试策略

提升了主板功能测试环节的整体经济效益，本文的主要创新工作总结如下：

(1) 提出了一种基于元件互连信息的主板可靠性分析方法。该方法建立了包含元件互连信息的故障树，基于故障树结构，分别针对主板维修数据充足和缺乏的两种场景，对主板进行可靠性分析，获得主板功能测试环节的可靠性指标。该方法能够在数据缺乏的情况下对主板的可靠性进行分析，并且充分利用了元件互连信息，进一步提升了主板可靠性分析的准确性。通过两组对比试验，验证了包含元件互连信息的故障树模型的准确性，同时验证了将可靠性分析结果用于测试策略设计的可行性。

(2) 提出一种融合决策树和可靠性分析的可解释测试策略设计方法。该方法借助前述的主板可靠性分析方法，融合具有强可解释性的决策树模型建模，能够提升主板功能测试环节设计的整体经济效益。通过对比实验，验证了融合决策树和可靠性分析的可解释测试策略设计方法的有效性。

## 未来展望

主板功能测试环节的优化对于电子制造业中经济效益的提升具有重大意义。本文提出了一套针对主板功能测试环节的可靠性分析方法和测试策略设计方法，但是上述研究均假设了未来时段数据和当前的现场数据具有很强的相似性。因此，具有一定的局限性，未来将从以下几方面展开研究：

(1) 目前有大量针对相关性的研究，例如将相关性用于特征选择从而提升主板故障诊断的效率或降低硬件开销、将测项相关性用于减少主板测试中的冗余测项等，上述研究均未将相关性用于未来测试结果的预测。由于测项相关性由主板结构决定，而本文所建立的故障树正是基于主板的结构信息设计的，所以未来可以通过故障树结构得到功能测项的相关性信息。

(2) 由于工业制造环境里必然存在工况的变化，导致未来时段数据发生较大变化，这时仅利用主板可靠性信息和现场测试结果难以设计出适合未来时段的测试策略。针对该问题，在未来工作中可以考虑指标动态变化信息，准确的预测出未来时段的良率，然后结合测项相关性信息，推算出未来数据的分布，据此设计出能够适应工况变化的主板功能测试策略。

## 参考文献

- [1] Barbe D. F.. Very Large Scale Integration (VLSI)[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982:178-221
- [2] Ye F., Zhang Z., Chakrabarty K., et al. Knowledge-Driven Board-Level Functional Fault Diagnosis[M]. Cham: Springer International Publishing, 2017:1-18
- [3] Xu T., Chakrabarty K.. Fault Modeling and Functional Test Methods for Digital Microfluidic Biochips[J]. IEEE Transactions on Biomedical Circuits and Systems, 2009, V3(4): 241-253
- [4] Vo T., Wang Z., Eaton T., et al. Design for Board and System Level Structural Test and Diagnosis[C]. IEEE International Test Conference, 2006: 1-10
- [5] Hoefflinger B.. ITRS: The International Technology Roadmap for Semiconductors[M]. Hoefflinger B. Chips 2020. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 161-174
- [6] Li Z., Colburn J. E., Pagalone V., et al. Test-cost optimization in a scan-compression architecture using support-vector regression[C]. VLSI Test Symposium (VTS), 2017: 1-6
- [7] Agrawal M., Chakrabarty K.. Test-Cost Modeling and Optimal Test-Flow Selection of 3-D-Stacked ICs[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, V34(9): 1523-1536
- [8] Zhang Z., Wang Z., Gu X., et al. Board-level fault diagnosis using Bayesian inference[C]. VLSI Test Symposium (VTS), 2010: 244-249
- [9] 张伟, 贾瑶. 系统可靠性分析方法与比较研究[J]. 环境技术, 2021, V39(04): 84-89
- [10] Shu M. H., Cheng C. H., Chang J. R.. Using intuitionistic fuzzy sets for fault-tree analysis on printed circuit board assembly[J]. Microelectronics Reliability, 2006, V46(12): 2139-2148
- [11] Wang J. Q.. New operators on triangular intuitionistic fuzzy numbers and their applications in system fault analysis[J]. Information Sciences, 2013, V251: 79-95
- [12] Yokogawa S., Kunii K.. Survey of critical failure events in on-chip interconnect by fault tree analysis[J]. Japanese Journal of Applied Physics, 2018, V57(7S2): 07MG01
- [13] Pan H.X., Huang J.Y., Liu G.M.. Fault diagnosis of circuit board based on fault tree[C]. Control, Automation, Robotics and Vision, 2008: 1666-1671
- [14] Shu X, Guo Y, Yang H, et al. Reliability study of motor controller in electric vehicle by the approach

- of fault tree analysis[J]. *Engineering Failure Analysis*, 2021, V121: 105165
- [15] Shu X., Guo Y., Yang W., et al. A Detailed Reliability Study of the Motor System in Pure Electric Vans by the Approach of Fault Tree Analysis[J]. *IEEE Access*, 2020: 5295-5307
- [16] Hedel J., Nguyen N., Abuelrub A.. Reliability Evaluation of Autonomous Electric Vehicles Using Fault Tree Method[C]. *North American Power Symposium (NAPS)*, 2023: 1-6
- [17] Akkouche N., Mir S., Simeu E., et al. Analog/RF test ordering in the early stages of production testing[C]. *VLSI Test Symposium (VTS)*, 2012: 25-30
- [18] Akkouche N., Bounceur A., Mir S., et al. Minimization of functional tests by statistical modelling of analogue circuits[C]. *Design & Technology of Integrated Systems in Nanoscale Era*, 2007: 35-40
- [19] Akkouche N., Mir S., Simeu E.. Ordering of analog specification tests based on parametric defect level estimation[C]. *VLSI Test Symposium (VTS)*, 2010: 301-306
- [20] Brockman J. B., Director S. W.. Predictive subset testing: optimizing IC parametric performance testing for quality, cost, and yield[J]. *IEEE Transactions on Semiconductor Manufacturing*, 1989, V2(3): 104-113
- [21] Milor L., Sangiovanni-Vincentelli A. L.. Minimizing production test time to detect faults in analog circuits[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1994, V13(6): 796-813
- [22] Han D., Halder A., Chatterjee A.. Test elimination using redundancy analysis for specification test of analog circuits[J]. *IEEE International Mixed Signal Testing Workshop*, 2004:420-425
- [23] Biswas S., Peng Li., Blanton R. D. S., et al. Specification Test Compaction for Analog Circuits and MEMS[C]. *Design, Automation and Test in Europe*. Munich, Germany: IEEE, 2005, V1: 164-169
- [24] Stratigopoulos H. G., Drineas P., Slamani M., et al. RF Specification Test Compaction Using Learning Machines[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2010, V18(6): 998-1002
- [25] Grady M., Pepper B., Patch J., et al. Adaptive testing - Cost reduction through test pattern sampling[C]. *IEEE International Test Conference (ITC)*, 2013: 1-8
- [26] Gonçalves H., Li X., Correia M., et al. A Fast Spatial Variation Modeling Algorithm for Efficient Test Cost Reduction of Analog/RF Circuits[C]. *Design, Automation & Test in Europe Conference &*

- Exhibition (DATE), 2015: 1042-1047
- [27] Ahmadi A., Nahar A., Orr B., et al. Wafer-level process variation-driven probe-test flow selection for test cost reduction in analog/RF ICs[C]. VLSI Test Symposium (VTS), 2016: 1-6
- [28] Wang K. L., Lin B. Y., Wu C. W., et al. Test Cost Reduction Methodology for In-FO Wafer-Level Chip-Scale Package[J]. IEEE Design & Test, 2016, V34(3): 50-58
- [29] Xue C., Shawn R. D.. A one-pass test-selection method for maximizing test coverage[C]. IEEE International Conference on Computer Design (ICCD), 2015: 621-628
- [30] Pan R., Zhang Z., Li X., et al. Black-Box Test-Coverage Analysis and Test-Cost Reduction Based on a Bayesian Network Model[C]. VLSI Test Symposium (VTS), 2019: 1-6
- [31] Liu M., Pan R., Ye F., et al. Fine-grained Adaptive Testing Based on Quality Prediction[J]. ACM Transactions on Design Automation of Electronic Systems, 2020, V25(5): 1-25
- [32] Mirhashemi M., Anvari R., Barari M., et al. Test-Cost Sensitive Ensemble of Classifiers Using Reinforcement Learning[J]. Revue d'Intelligence Artificielle, 2020, V34(2): 143-150
- [33] Feige U.. A threshold of  $\ln n$  for approximating set cover[J]. Journal of the ACM, 1998, V45(4): 634-652
- [34] Pan R., Zhang Z., Li X., et al. Black-Box Test-Cost Reduction Based on Bayesian Network Models[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, V40(2): 386-399
- [35] Pan R., Zhang Z., Li X., et al. Black-Box Test-Cost Reduction Based on Bayesian Network Models[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, V40(2): 386-399
- [36] Barlow R. E., Proschan F. Importance of system components and fault tree events[J]. Stochastic Processes and their Applications, 1975, V3(2): 153-173
- [37] Hu Y. N. Research on the Application of Fault Tree Analysis for Building Fire Safety of Hotels[J]. Procedia Engineering, 2016, 135: 524-530
- [38] Ciallella H. L., Zhu H.. Advancing Computational Toxicology in the Big Data Era by Artificial Intelligence: Data-Driven and Mechanism-Driven Modeling for Chemical Toxicity[J]. Chemical Research in Toxicology, 2019, V32(4): 536-547

- [39] Zhao L., Russo D. P., Wang W., et al. Mechanism-Driven Read-Across of Chemical Hepatotoxicants Based on Chemical Structures and Biological Data[J]. *Toxicological Sciences*, 2020, V174(2): 178-188
- [40] Ni Y., Xu J., Zhu C., et al. Accurate residual capacity estimation of retired LiFePO<sub>4</sub> batteries based on mechanism and data-driven model[J]. *Applied Energy*, 2022, V305: 117922
- [41] Yoo S. H., Geng H., Chiu T. L., et al. Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging[J]. *Frontiers in medicine*, 2020, V7:427
- [42] Rokach L., Maimon O.. Top-Down Induction of Decision Trees Classifiers—A Survey[J]. *IEEE Transactions on Systems*, 2005, V35(4): 476-487
- [43] Li Q., Bermak A.. A Low-Power Hardware-Friendly Binary Decision Tree Classifier for Gas Identification[J]. *Journal of Low Power Electronics and Applications*, 2011, V1(1): 45-58
- [44] Song Y. Y., Lu Y.. Decision tree methods: applications for classification and prediction[J]. *Shanghai Arch Psychiatry*, 2015, V27(2): 130-5
- [45] Petre, E. G.. A decision tree for weather prediction. *Bulletin of PG University of Ploiesti, Series Mathematics, Informatics, Physics*, 2009, V61(1):77-82.
- [46] 张俊玉, 胡家豪, 黄言, 等. CART 决策树方法在煤电厂节能降耗中的应用[J]. *控制与决策*, 2021, V36(05): 1232-1238.
- [47] Ye F., Zhang Z., Chakrabarty K., et al. Adaptive Board-Level Functional Fault Diagnosis Using Decision Trees[C]. *Asian Test Symposium*, 2012: 202-207
- [48] Ye F., Zhang Z., Chakrabarty K., et al. Adaptive Board-Level Functional Fault Diagnosis Using Incremental Decision Trees[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016, V35(2): 323-336
- [49] Rathore S. S., Kumar S.. A Decision Tree Regression based Approach for the Number of Software Faults Prediction[J]. *ACM SIGSOFT Software Engineering Notes*, 2016, V41(1): 1-6
- [50] Rathore S. S., Kumar S. An empirical study of some software fault prediction techniques for the number of faults prediction[J]. *Soft Computing*, 2017, V21(24): 7417-7434
- [51] Rathore S. S., Kumar S.. A study on software fault prediction techniques[J]. *Artificial Intelligence Review*, 2019, V51(2): 255-327

- 
- [52] Biswas S., Blanton R. D.. Test Compaction for Mixed-Signal Circuits Using Pass-Fail Test Data[C]. VLSI Test Symposium, 2008: 299-308
- [53] Biswas S., Wang H., Blanton R. D.. Reducing test cost of integrated, heterogeneous systems using pass-fail test data analysis[J]. ACM Transactions on Design Automation of Electronic Systems, 2014, V19(2): 1-23
- [54] 陈嘉慧, 米洁, 张胜伦, 等. 基于故障树和贝叶斯网络的履带车辆行星变速机构可靠性分析[J]. 机械强度, 2024, V46(01): 129-135
- [55] Wang H., Liu L., Wang Y., et al. An Overview of Failure Analysis Expert System Based on Machine Learning[C]. International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2019: 584-589
- [56] Jian W. X., Futatsugi K., Yan X. H.. Fault tree and formal methods in system safety analysis[C]. The Fourth International Conference on Computer and Information Technology, 2004: 1108-1115
- [57] Hong X., Dugan J. B.. Combining dynamic fault trees and event trees for probabilistic risk assessment[C]. Annual Symposium Reliability and Maintainability, 2004: 214-219
- [58] 李泓波, 白劲波, 杨高明等. 决策树技术研究综述[J]. 电脑知识与技术, 2015, V11(24): 1-4
- [59] Kotsiantis S. B.. Decision trees: a recent overview[J]. Artificial Intelligence Review, 2013, V39(4): 261-283
- [60] Aitkenhead M. J.. A co-evolving decision tree classification method[J]. Expert Systems with Applications, 2008, V34(1): 18-25
- [61] Blockeel H., Devos L., Frénay B., et al. Decision trees: from efficient prediction to responsible AI[J]. Frontiers in Artificial Intelligence, 2023, 6: 1124553
- [62] Hassan A. M., Ameer A. H., Hossain M. I., et al. A Novel Pessimistic Decision Tree Pruning Approach for Classification[C]. Electrical Information and Communication Technology (EICT), 2023: 1-6
- [63] Manzali Y., Mohamed E. F.. A new decision tree pre-pruning method based on nodes probabilities[C]. Intelligent Systems and Computer Vision (ISCV), 2022: 1-5
- [64] Xie H., Shang F.. The study of methods for post-pruning decision trees based on comprehensive evaluation standard[C]. Fuzzy Systems and Knowledge Discovery (FSKD), 2014: 903-908
- [65] Zhang W., Li Y.. A Post-Pruning Decision Tree Algorithm Based on Bayesian[C]. Computational and

Information Sciences, 2013: 988-991

- [66] Gilpin L. H., Bau D., Yuan B. Z., et al. Explaining Explanations: An Overview of Interpretability of Machine Learning[C]. Data Science and Advanced Analytics (DSAA), 2018: 80-89
- [67] Liu B., Zhao Y., Kang Y., et al. A Feature Engineering-based Method for PCB Solder Paste Position Offset Prediction[J]. International Symposium on Autonomous Systems, 2023: 1-6
- [68] Corsi F., Martino S., Williams T. W.. Defect level as a function of fault coverage and yield[C]. Proceedings ETC 93 Third European Test Conference, 1993: 507-508

## 攻读硕士学位期间取得的学术成果

- [1]. Li Y, Wang K, Kang Y, et al. Board-level Functional Test Selection Based on Fault Tree Analysis[C]. International Symposium on Autonomous Systems (ISAS), 2023: 1-6
- [2]. 赵云波, 李瑶瑶, 王康成等. 一种基于可靠性分析的主板功能测试策略方法及系统[P]. 安徽省: CN116541218A, 2023-08-04
- [3]. 赵云波, 李瑶瑶, 王康成等. 一种细粒度自适应主板功能测项选择方法及系统[P]. 安徽省: CN116594816A, 2023-08-15
- [4]. 赵云波, 李瑶瑶, 王康成等. 融合决策树和故障树分析的主板功能测试策略设计方法[P]. 安徽省: CN117389808A, 2024-01-12

## 攻读硕士学位期间参与的科研项目

- [1]. 非可信智能驱动的可靠智造（安徽省发改委重大专项，No.21KT010）
- [2]. 人机智能协同关键技术及其在智能制造中的应用（安徽省重点研发计划面上攻关项目，No.202104a05020064）

## 致 谢

研究生三年的时光如白驹过隙，回首这段旅程，我满载而归，感慨良多。从初入学术殿堂的迷茫，到如今即将毕业的自信与从容，每一步成长都离不开众多贵人的帮助。

在学业上，我深入研究了专业领域的知识，不断挑战自己的认知边界，不仅积累了丰富的理论知识，更通过实践培养了解决问题的能力。在思想上，我更加开放和成熟，学会了多角度看待问题，对待学术和生活都有了更深刻的理解。

我要特别感谢在我的研究生阶段给予我莫大帮助的三位老师。赵云波老师的和蔼可亲、儒雅风范总是让我如沐春风，他在人机协同、智能制造领域的独到见解更是让我受益匪浅。我还要感谢陈思宝老师，他的责任心和专业精神让我深感敬佩。尽管工作繁忙，学生众多，陈老师始终坚持每周组织论文阅读，每天分享高质量论文，为实验室营造了浓厚的学习氛围。王康成老师不仅是一位严谨、认真、细心、友善的科研工作者，更是一位尽职尽责、充满智慧的良师。在他的指导下，我逐渐树立了正确的科研思维，学会了如何进行科学实验、收集和分析数据、撰写学术论文等科研方法。每当我遇到困惑或挑战时，王老师总是耐心地为我解答疑难问题，指导我修改论文，帮助我顺利通过开题答辩。王老师的严谨治学态度和深厚的学术造诣让我深受启发，我的每一点进步都凝聚着王老师的心血和智慧。

感谢实验室伙伴们的陪伴与支持。与他们一起学习，共同探讨学术问题，不仅让我收获了知识，更让我感受到了团队的力量和友谊的温暖。在科研过程中，我们相互帮助，共同进步，他们的陪伴让枯燥的实验变得有趣且充满动力。感谢实验室伙伴们为我的科研之旅增添了无尽的乐趣和回忆。

感谢我的室友王璐、王怡和王若彤。三位来自北方的姑娘用她们的情感和豁达感染了我，让我更加深入地体验了北方的风土人情。我们在一起的时光总是充满欢声笑语，她们为我的日常生活增添了无数幸福时刻。

家人和朋友一直是我坚实的后盾。父母的爱与支持是我前进的动力源泉，无论我遇到什么困难或挑战，他们总是第一时间给予我鼓励和支持。感谢我的好朋友们严亚军、朱彬彬、黄洁、汪鑫、章炫玉、余婷、汪峰和方航一直陪伴在我身边，在我需要的时候伸出援手。同时，我也要感谢在研究生期间认识的男朋友，他的出现为我的生活带来了色彩和温暖，是我在这段时间里收到的最美好的礼物。

在毕业论文的完成过程中，我得到了开题答辩、中期答辩以及毕业预答辩中各位老师的宝贵建议。他们提出的建设性意见不仅丰富了我的论文内容，更提升了我的学术水平。在此向各位老师表示衷心的感谢。

三年的研究生生活即将画上句号，但这段经历将永远铭刻在我的心中。感谢所有帮助过我的人，在未来的日子里，我将带着这份感恩之心继续前行，不断探索和进步。