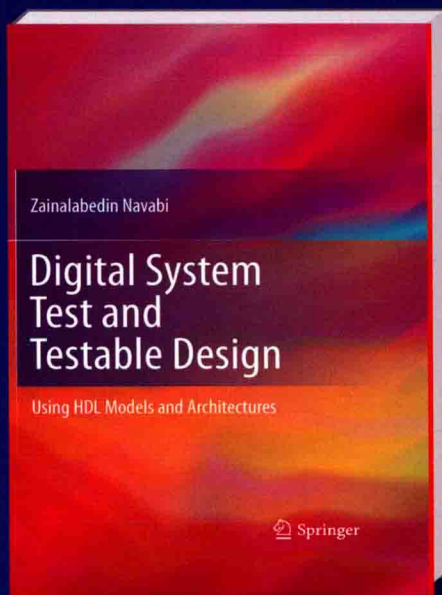


由浅入深，从测试的基本概念，深入到DFT的原理、电路和验证实例。  
体系完整，从如何解决测试问题的角度出发，覆盖了DFT领域的所有环节，面面俱到。  
案例实用，书中案例具有工程参考价值，部分改进后可被用于搭建测试电路。



电子与嵌入式系统  
设计译丛



Digital System Test and Testable Design  
Using HDL Models and Architectures

# 数字系统测试和 可测试性设计

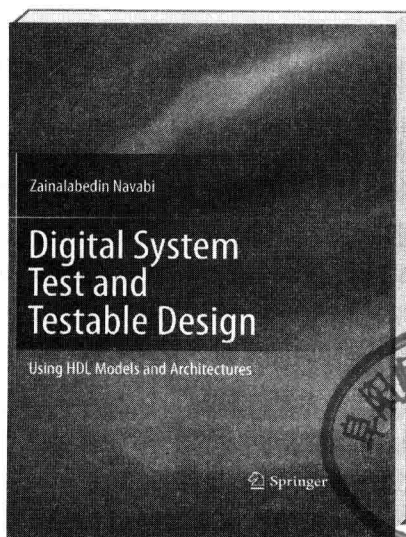
[美] 塞纳拉伯丁·纳瓦比 (Zainalabedin Navabi) 著  
贺海文 唐威昀 译



机械工业出版社  
China Machine Press



电子与嵌入式系统  
设计译丛



Digital System Test and Testable Design  
Using HDL Models and Architectures

# 数字系统测试和 可测试性设计

[美] 塞纳拉伯丁·纳瓦比 (Zainalabedin Navabi) 著  
贺海文 唐威昀 译



机械工业出版社  
China Machine Press

## 图书在版编目 ( CIP ) 数据

数字系统测试和可测试性设计 / (美) 纳瓦比 (Navabi, Z.) 著; 贺海文, 唐威均译. —北京: 机械工业出版社, 2015.4

(电子与嵌入式系统设计译丛)

书名原文: Digital System Test and Testable Design: Using HDL Models And Architectures

ISBN 978-7-111-50154-1

I. 数… II. ① 纳… ② 贺… ③ 唐… III. 数字系统—系统测试 IV. TP271

中国版本图书馆 CIP 数据核字 (2015) 第 094119 号

---

本书版权登记号: 图字: 01-2013-6797

Translation from the English language edition:

Digital System Test and Testable Design: Using HDL Models and Architectures by Zainalabedin Navabi, Copyright © 2011 Springer US.

Springer US is a part of Springer Science+ Business Media.

All Rights Reserved.

本书中文简体字版由 Springer Science+Business Media 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

## 数字系统测试和可测试性设计

---

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张梦玲

责任校对: 董纪丽

印 刷: 三河市宏图印务有限公司

版 次: 2015 年 6 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 24

书 号: ISBN 978-7-111-50154-1

定 价: 85.00 元

---

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzlit@hzbook.com

版权所有 · 侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

# 译者序

近些年来，随着集成电路设计和制造技术的高速发展，集成电路的测试成为了一个越来越重要的问题。在实际的集成电路设计流程中，可测试性设计的理论和相关实现技术是我们必须特别重视的设计要素，可测试性很差的设计将无法量产。因此，在这个领域，每年都有很多书籍或论文出版或发表。Zainalabedin Navabi 教授的这本书就是该领域内一部比较全面和优秀的著作。本书的最大特点在于理论和实践很好地结合，通过使用 Verilog 编写的数字电路实例、测试应用实例和测试平台，读者在学习可测试性技术的同时还能学习 Verilog 语言。同时通过测试平台的编译和仿真、过程语言接口 (PLI) 的使用，使读者更加深入地理解和掌握这些知识。

本书系统地介绍了数字系统的电路测试以及可测试性设计、使用 Verilog 来进行数字电路的设计、测试分析和实现数字系统的可测试性设计。本书也是根据作者多年的科研成果和教学实践，结合集成电路可测试性设计关注的基本理论和研究热点，并参考大量的文献撰写而成的。全书共 11 章，第 1 章~第 4 章介绍数字电路测试和 Verilog RTL 设计基础知识，包括测试的基本概念、测试设备、测试经济学和故障模型等，以及用 Verilog 来建立数字电路测试平台和仿真的概念。第 5 章、第 6 章介绍测试方法，包括数字电路测试向量生成方法、确定性数字电路测试生成方法。第 7 章~第 11 章介绍可测试性设计方法，包括扫描设计、边界扫描测试、逻辑内建自测试、数字电路测试压缩、MBIST (存储器内建自测试)。

本书是首本全面覆盖 Verilog 语言使用、数字电路设计、数字电路测试、存储器测试、测试平台仿真的专著和教科书，反映了当今集成电路测试的研究现状和发展趋势，可以作为高等学校计算机、微电子、电子工程等相关专业高年级学生和研究生教材与参考书，也可以供从事上述领域工作的科研技术人员参考。

参加本书翻译的还有唐威昫工程师，他为此付出了很多心血和时间。在本书的翻译过程中，也一直得到机械工业出版社华章公司张国强老师的热心指导和大力支持，他向我们推荐了这本著作，并帮助我们与作者进行沟通。本人谨向为本书的翻译与出版付出辛勤劳动的各位编辑，以及支持我们进行翻译工作的家人和朋友，致以衷心的感谢。

由于译者水平有限，书中难免有一些错误或者不妥之处，真诚希望各位读者在阅读时把发现的错误及时告知我们，多多批评指正。我们的个人邮箱是：hehaiwen2005@sina.com, tangweiyun@tsinghua.org.cn。

# 前 言

这是一本关于测试和数字电路可测试性的书。在这本书中，使用系统设计的语言来介绍和设计测试电路。在介绍数字电路设计的实践和方法的同时，介绍了测试和可测试性的概念。在本书中，我们展示了如何测试数字电路，如何设计可测电路，如何充分使用现有的寄存器传输级（RTL）设计、验证方法和工具。

本书采用 Verilog 模型和 Verilog 测试平台，用它们实现故障仿真和测试生成算法，并解释仿真和测试结果。在介绍可测电路的部分，我们用 Verilog 描述了各种扫描和 BIST 方法。同时，使用 Verilog 的测试平台搭建虚拟测试机，以检查和评估这些可测试性方法。在设计可测试性电路的过程中，使用 Verilog 的测试平台来评估这些电路，从而提高主设计的可测试性。

首先，我们建立了 Verilog 测试环境，它可以进行门级故障仿真和测试生成。同时使用 Verilog PLI 与功能强大的 Verilog 测试平台进行硬件建模并编写测试环境。

接着，我们采用 Verilog 作为硬件设计工具，用它来描述可测试性设计（DFT）和 BIST 硬件。在这一部分，我们使用 Verilog 作为硬件描述语言，描述和综合产生可测试硬件电路。在书中，我们还使用 Verilog 建模和仿真，这有助于制定和评估测试方法和测试电路的硬件结构。

本书提供了一个新的教授测试以及测试电路的方法，与其他介绍测试和可测试性的书籍相比较，这本书的特点是使用了 Verilog 和 Verilog PLI 测试的应用程序。

在 20 世纪 70 年代后期，硬件描述语言（HDL）用于计算机体系结构的教授中。在现在，HDL 也可以用来阐明测试方法和可测电路的架构。否则，我们只能通过非正式的流程图和框图来表述这些内容。

使用 Verilog 语言的好处有：

第一，消除了测试算法、BIST 以及 DFT 硬件电路架构中的模糊情况，它能够清楚地描述可测试性的硬件、体系结构以及测试场景。

第二，用 Verilog 描述片上测试硬件，这有助于评估相关算法、硬件开销和测试时间，并以此评估在 SoC 芯片上使用这些片上测试硬件的可行性。

第三，Verilog 语言也可以支持我们使用测试平台。在开发测试平台和虚拟测试机时，PLI 为我们提供了强大的编程工具。这个工具可以与硬件（用 Verilog 语言描述）进行交互。

这种混合的硬件 / 软件环境将有利于我们描述复杂的测试方案和测试策略。

## 本书主要内容

### 数字电路测试的基础知识和 HDL 的作用

本章介绍数字电路测试的基础知识，讨论了数字系统测试的重要性，并且定义了各种数字电路测试专用术语，还讨论了数字电路测试的成本及减少数字电路测试时间的途径，这些途径包括更好的数字电路测试方法、更多的可测试性设计、更有效率的测试。另外，本章还探讨了数字电路设计和数字电路测试的关系。

### 用于设计和测试的 Verilog HDL

本章介绍了用于描述数字电路系统和相应测试平台的 Verilog 硬件描述语言。我们使用 Verilog 进行组合逻辑以及时序逻辑电路建模，同时还给出了几个示例。在这里，我们只介绍本书要用到的关键语言结构，这些结构可以用于理解电路模型以及电路体系结构。本章展示了如何使用 Verilog 开发良好的设计测试平台。我们讨论了自己开发的几个测试平台模板，在开发这些测试平台过程中，我们展示了 PLI 和开发出来的 PLI 函数的使用。在后面的章节中我们将继续介绍测试平台。

### 故障和缺陷建模

本章首先描述了晶体管级和门级故障，用我们的 Verilog 仿真说明了底层晶体管级故障和上层门级故障之间的对应关系，并论述了功能性和结构性故障，以及它们之间的区别，还讨论了结构性门级故障，同时使用模拟仿真证明这种模型的合理性。另外，本章还阐述了固定 (Stuck-at) 故障模型，同时介绍了如何使用 PLI 函数进行故障注入。接下来，本章论述了故障等效和几种故障压缩技术，并开发了基于 Verilog 的 PLI 函数和测试平台，用它们产生故障列表，进行故障压缩，接下来还用这些测试平台测试了几种基准电路。

### 故障仿真应用与方法

本章首先论述了故障仿真的用途及其在数字电路设计与测试中的应用。其次论述了不同的故障仿真技术，包括串行、并行、并发、演绎、微分和关键路径，用它们来追踪故障仿真。为了实现这些方法，我们开发了一个测试平台以进行固定故障注入和仿真，这个基于 PLI 的 Verilog 测试平台也部分实现了其他的故障仿真技术。再次，我们使用基于 PLI 的 Verilog 测试平台产生了故障字典。基于这些工具的使用，本章讨论了数字电路测试覆盖率及其实现、如何减少故障和其他与故障仿真相关的概念。在第 3 章中我们使用基于 PLI 的 Verilog 测试平台产生了故障表，本章使用相同的故障表格式。在本章中开发和使用了几个完整的基于 PLI 的 Verilog 测试平台。

## 测试向量生成方法及算法

本章首先介绍了各种可测试性技术，包括基于概率、结构和 SCOPE 参数的计算技术。我们开发了基于 PLI 的 Verilog 测试平台，用于计算门级电路内部节点的可控性和可观察性参数。其次讨论了可检测性和它在确定随机测试方法中的作用。再次，我们论述了各种随机数字电路测试生成方法，这一部分还涉及如何利用前面介绍的可测试性测试方法。本章使用 Verilog 测试平台生成随机测试，并且用它进行基于 Verilog 的故障仿真来评估这些随机测试。

## 确定性测试生成算法

在第 5 章中，我们介绍了数字电路测试向量生成中的随机数字电路测试生成方法。本章具体讨论确定性数字电路测试生成，这是我们考虑测试生成流程的第二阶段。我们论述了算法，例如，*D* 算法、PODEM、CPT 和这些算法的简化衍生。我们开发完成了 Verilog 测试平台，这个平台使用了 Verilog PLI 函数，用于确定何时停止随机数字电路测试生成以及决定何时开始确定性数字电路测试生成。数字电路测试生成的下一阶段是数字电路测试压缩。本章有一部分内容专门讨论了这个论题，以及相关的几种数字电路测试压缩方法，同时我们也介绍了如何使用 Verilog 实现这些内容。

## 扫描设计方法——可测试性设计方法之一

在本章和接下来的章节中，除了在开发数字电路测试环境中使用 Verilog，也可以用 Verilog 描述实际的硬件结构。本章介绍了可测试性设计，展示了所提出的可测试性设计体系结构的可综合 Verilog 代码。本章首先介绍了几个特定可测试性设计技术，其次介绍了全扫描和各种部分扫描体系结构，以及对这些硬件结构的明确描述，同时展示了相应的 Verilog 代码。我们在本书前面的章节中介绍了几种数字电路测试方法以及测试平台。在这里，我们可以利用它们进行扫描设计评估和改进。本章展示了如何使用一个测试平台来配置扫描设计电路以及产生这个电路的测试向量，还说明了如何使用一个 Verilog PLI 测试平台，同时将这个平台用于基于扫描的数字设计的应用与测试，后者所应用的测试平台就是我们所说的虚拟测试机。

## 标准 IEEE 测试访问方法——可测试性设计方法之二

本章论述了 IEEE 1149.1 数字电路测试标准，还论述了与这些硬件结构对应的 Verilog 实现。虚拟测试机包括操作板和核心测试硬件，也被描述为 Verilog 的测试平台。通过使用 Verilog，我们能够展示这个架构和这些标准的应用。本章描述了支持 IEEE 1149.1 标准的各个部件的接口，也说明了如何实现该标准与被测电路的交互，以及如何实现测试设备与被测电路交互。同时，在这里要澄清的是，以上都是使用 Verilog 进行硬件描述和测试平台来实现的。

## 逻辑内建自测试——可测试性设计方法之三

本章开头介绍了芯片上数字电路测试数据生成电路的设计和输出分析的方法。接下来，我们把这些电路组件组装起来，作为内建自测试架构进行片上测试。本章展示了可综合的 Verilog 代码，这些代码可用于我们介绍的所有内建自测试电路架构，还展示了经典的内建自测试电路架构，例如随机测试接口（RTS）、内建逻辑模块观察器（BILBO）、内建评估和自检（BEST）。此外还有在线内建自测试、并发内建自测试和特殊体系结构所使用的内建自测试。通过 Verilog，我们可以清楚地描述这些硬件结构。同时，可以在使用的 Verilog 以及 PLI 测试平台上进行故障注入和故障仿真，对内建自测试电路进行配置和评测。通过 Verilog 仿真测定哪一种内建自测试电路有更高的故障覆盖率，同时确定这种电路具有的特征。

## 测试压缩

本章论述了数字电路测试压缩技术，该技术用于可测试性设计以及相应的硬件实现。我们讨论了 Huffman、Run-length、Golomb，以及用于测试压缩的其他编码技术。此外，我们讨论了扫描压缩技术及其相应的芯片上的扫描结构。同时还介绍了扫描压缩的压缩算法以及在芯片上的解压缩硬件（用 Verilog 描述）。

## 通过 MBIST 测试存储器——可测试性设计方法之四

本章首先介绍了存储器结构以及相应的故障模型。对各种 March 数字电路测试技术进行了介绍和分析，证明了不同测试算法的合理性。然后，我们讨论了几个 MBIST 架构，并给出了相应的 Verilog 描述。本章通过一个 Verilog 测试平台演示了如何进行存储器自测试。

## 附录

- 附录 A 在协议感知自动测试设备上使用 HDL
- 附录 B PLI 测试应用的门组件
- 附录 C 编程语言接口测试工具
- 附录 D IEEE 1149.1 标准边界扫描的 Verilog 描述
- 附录 E 边界扫描 IEEE 1149.1 标准虚拟测试机
- 附录 F 由 RTL 综合生成的门级网表（NetlistGen）

## 本书相关软件和课程资料

作者在几所大学教授数字电路测试和可测试性课程期间，已经完成了本书所有的相关资料，且可以提供这些资料。

作者制定了一整套用于数字电路测试开发的 Verilog PLI 函数，这些函数已应用于几门课程的教学，并达到了一个可以接受的成熟度。为了使数字电路测试应用能够使用原始的



RTL 设计，我们开发了一个软件，这个软件使用 Xilinx ISE 来综合、转换以产生网表，并供 PLI 函数使用。本书使用的许多例子都是由本软件生成的网表。书中的应用程序可以执行我们的 PLI 函数以及网表。这些应用程序包括故障压缩、随机数字电路测试生成、故障仿真，以及可测试性测量评估。同时，本书还讨论了使用网表和 PLI 函数仿真的虚拟测试机。

我们还可提供所有章节的演示材料（PowerPoint 幻灯片）。另外，在教授这门课程时，作者注意到，学生有时需要回顾 Verilog 或数字逻辑设计概念，我所制作的视频短片对此有复习、指导作用，学生可以在它的指导下准备书中所需的资料。视频和软件的使用手册也可以从作者处获得，作者邮箱：navabi@ece.wpi.edu; zain@navabi.com。

在本书出版的时候，相关配套资料可从施普林格亚洲有限公司北京代表处获取，联系电话：010-82670211。

# 概 述

本书主要论述了数字系统的电路测试以及设计可测试性。本书使用 Verilog 进行数字电路的设计、测试分析并实现数字系统可测试性。在第 1 章中，我们讨论了数字电路测试和可测试性设计的基础知识，同时也论述了使用 HDL 实现数字电路测试。这一部分详细描述了整个数字系统测试，包括数字电路测试方法、数字电路可测试性的实现和检验方法，同时还讨论了如何使用 HDL 来实现以上三方面的内容。在介绍完这些内容之后，我们专门用一章内容介绍了 Verilog 基础知识以及如何使用这种语言进行数字电路设计和测试。本书的主要内容都是基于 Verilog RTL 设计的基础知识，并且用它来建立数字电路测试和仿真。

从第 3 章开始，本书的重点回到了数字电路测试问题上，例如故障压缩、故障仿真，以及数字电路测试向量生成。从第 3 章开始，有 4 章内容涉及“数字电路测试方法”，第 3 章介绍故障模型，第 4 章介绍故障仿真方法，随后的两章介绍数字电路测试向量生成，讨论随机数字电路测试生成（基于 HDL）和确定性数字电路测试生成。

为了实现这些应用，我们使用 Verilog 门级模型以及基于 PLI 的测试平台，这些模型和平台能够支持故障注入，进行故障仿真以及数字电路测试向量生成。在这一部分中，Verilog 的测试平台作为数字电路测试程序，可以用于管理电路的结构模型，这个模型可用于进行故障仿真、计算故障覆盖率和数字电路测试向量生成。

本书从第 7 章开始介绍可测试性设计，这一部分有 4 章（第 7～10 章），介绍了不同的可测试性设计方法、内建数字电路自测试体系结构，以及数字电路测试压缩方法。我们使用 Verilog 编写代码来描述不同的可测试性设计方法以及内建自测试体系结构的硬件实现。在这一部分，我们搭建了 Verilog 测试平台，它可以作为虚拟测试机，用来检测嵌入电路设计中的可测试性设计电路硬件及其可测试性。同时，第 3～6 章论述了基于 PLI 故障仿真以及数字电路测试向量生成的 Verilog 环境，并且用 Verilog 编码搭建了可测试性硬件，这些硬件提供了一个完整的环境，用于数字电路的可测试性和内建自测试评测。通过以上的评测，可以帮助我们实现以及完善数字电路测试硬件设计。这一部分也描述了边界扫描的 IEEE 标准以及内核测试方法，同时也告诉我们如何使用 Verilog 描述这些标准以及使用这些标准进行数字电路设计。最后一章是存储器测试，重点讨论了 MBIST（采用 Verilog 描述）。

# 致 谢

在德黑兰大学的数字系统测试课程中，我想到使用一种 HDL 来搭建和描述测试电路。当时，我使用的是所授课程中的 VHDL 进行测试。经过几年的教学经验总结，我们切换到 Verilog 语言并且开发了相应的一套工具库（它就是有助于实现这些测试功能的 Verilog 组件）。开发这些软件并帮助我完成一整套相关课程材料收集的学生对本书的写作有非常重要的贡献。在此特别感谢负责 Verilog 软件包开发的 Nastaran Nemati，她在读研究生阶段（2010 年）负责整个工具库研发的管理工作，对此项工作贡献巨大。同时我也感谢来自美国马萨诸塞州伍斯特理工学院和德黑兰大学的学生，他们对此课程提出了很多有益的建议。

在这本书的实际写作过程中，我带的研究生 Fatemeh (Negin) Javaheri 是和我一起讨论想法的关键人物。基于本书的内容，她总是与我进行探讨，她的想法对构建本书结构也有显著的帮助。后来，她负责了本书测试压缩章节的素材开发。Negin 现继续在我的研究领域里和我一起工作，同时也很期待这些工作成果能够出现在我的下一本书中。另外一个有重要贡献的人是 Somayeh Sadeghi Kohan（也是德黑兰大学的研究生），Somayeh 为本书中边界扫描章节开发了素材。在本书成稿工作的最后阶段，她对本书的章节审校和修改有很大贡献，同时她所提供的反馈和修改建议非常有用。Nastaran Nemati 协助写作了 HDL 章节，Parisa Kabiri 和 Atie Lotfi 也对一些章节的审校和写作有所贡献。

与往常我写书一样，Fatemeh Asgari 负责管理这个项目（在过去 20 年中她一直是我的助手），她负责管理那些参与研究、开发软件、收集素材、为终稿准备文字和插图的学生。Fatemeh 的大力支持和出众的管理能力一直是我写作和项目研究能得以成功的关键因素。非常感谢她这些年来为我做的工作。

出于我的工作习惯和时间安排，我不得不离开我的家人，进行我的研究和写作，这对他们来说是一件难事。不过，他们一直支持和理解我，并鼓励我完成所有的项目。我的妻子 Irma 一直对我有很大的帮助，使我可以安心地花很多时间进行写作。感谢 Irma，还有我的儿子 Aarash 和 Arvand。

Zainalabedin Navabi  
(navabi@ece.wpi.edu)

# 目 录

译者序  
前 言  
概 述  
致 谢

## 第 1 章 数字电路测试的基础知识和 HDL 的作用 / 1

- 1.1 设计及测试 / 1
  - 1.1.1 RTL 设计流程 / 1
  - 1.1.2 流片后测试 / 4
- 1.2 测试重点 / 7
  - 1.2.1 测试方法 / 7
  - 1.2.2 可测试性方法 / 9
  - 1.2.3 检测方法 / 11
  - 1.2.4 测试成本 / 11
- 1.3 数字系统测试中的 HDL / 13
  - 1.3.1 硬件建模 / 13
  - 1.3.2 制定测试方法 / 13
  - 1.3.3 虚拟测试机 / 14
  - 1.3.4 可测试性硬件评估 / 14
  - 1.3.5 协议感知自动测试设备 / 14
- 1.4 自动测试设备结构及仪器 / 14
  - 1.4.1 数字激励及测量仪器 / 15
  - 1.4.2 DC 仪器 / 15
  - 1.4.3 AC 仪器 / 15
  - 1.4.4 RF 仪器 / 15
  - 1.4.5 自动测试设备 / 16
- 1.5 小结 / 17

## 第 2 章 用于设计和测试的 Verilog HDL / 18

- 2.1 使用 HDL 开发测试方法的原因 / 18
- 2.2 将 Verilog 用于设计 / 19
  - 2.2.1 将 Verilog 用于仿真 / 19
  - 2.2.2 将 Verilog 用于综合 / 19
- 2.3 将 Verilog 用于测试 / 20
  - 2.3.1 无故障电路分析 / 21
  - 2.3.2 故障表编制及可测试性分析 / 21
  - 2.3.3 故障仿真 / 21
  - 2.3.4 测试生成 / 22
  - 2.3.5 可测试性硬件设计 / 22
- 2.4 Verilog 的基本结构 / 23
  - 2.4.1 模块、端口、连线及变量 / 24
  - 2.4.2 抽象的层级 / 25
  - 2.4.3 逻辑值系统 / 25
- 2.5 组合电路 / 26
  - 2.5.1 晶体管级描述 / 26
  - 2.5.2 门级描述 / 26
  - 2.5.3 运算级描述 / 27
  - 2.5.4 过程级描述 / 28
  - 2.5.5 实例化其他模块 / 29
- 2.6 时序电路 / 30
  - 2.6.1 寄存器和移位寄存器 / 31
  - 2.6.2 状态机编码 / 31
- 2.7 完整示例 (加法器) / 35
  - 2.7.1 控制 / 数据划分 / 35
  - 2.7.2 加法器的设计规格 / 36

- 2.7.3 CPU 的实现 / 36
- 2.8 测试平台技术 / 40
  - 2.8.1 测试平台技术 / 41
  - 2.8.2 简单的组合测试平台 / 41
  - 2.8.3 简单的时序测试平台 / 42
  - 2.8.4 限制数据集 / 43
  - 2.8.5 同步数据和响应处理 / 44
  - 2.8.6 随机时间间隔 / 45
  - 2.8.7 文本 IO / 45
  - 2.8.8 仿真代码覆盖率 / 47
- 2.9 PLI 基础知识 / 48
  - 2.9.1 访问例行程序 / 49
  - 2.9.2 HDL/PLI 实现的步骤 / 49
  - 2.9.3 在 HDL/PLI 环境中注入故障 / 51
- 2.10 小结 / 54
- 第 3 章 故障和缺陷建模 / 55**
  - 3.1 故障建模 / 55
    - 3.1.1 故障抽象 / 56
    - 3.1.2 功能故障 / 58
    - 3.1.3 结构故障 / 58
  - 3.2 门级结构故障 / 60
    - 3.2.1 确认故障 / 60
    - 3.2.2 固定开路故障 / 61
    - 3.2.3 固定为 0 的故障 / 62
    - 3.2.4 固定为 1 的故障 / 62
    - 3.2.5 桥接故障 / 62
    - 3.2.6 状态依赖型故障 / 63
    - 3.2.7 多故障 / 64
    - 3.2.8 单固定结构故障 / 64
    - 3.2.9 检测单固定故障 / 70
  - 3.3 与门级故障相关的问题 / 71
    - 3.3.1 检测桥接故障 / 71
    - 3.3.2 不可检测的故障 / 72
    - 3.3.3 冗余故障 / 72
- 3.4 故障压缩 / 72
  - 3.4.1 难以区分的故障 / 72
  - 3.4.2 等效单固定故障 / 73
  - 3.4.3 面向门的故障压缩 / 74
  - 3.4.4 面向线路的故障压缩 / 75
  - 3.4.5 重汇聚扇出的问题 / 76
  - 3.4.6 支配性故障压缩 / 76
- 3.5 基于 Verilog 的故障压缩 / 78
  - 3.5.1 用于故障压缩的 Verilog 测试平台 / 78
  - 3.5.2 故障压缩的 PLI 实现 / 79
- 3.6 小结 / 83
- 第 4 章 故障仿真应用与方法 / 84**
  - 4.1 故障仿真 / 84
    - 4.1.1 门级故障仿真 / 84
    - 4.1.2 故障仿真要求 / 85
    - 4.1.3 HDL 环境 / 86
    - 4.1.4 时序电路故障仿真 / 90
    - 4.1.5 故障排除 / 91
    - 4.1.6 相关术语 / 91
  - 4.2 故障仿真应用 / 92
    - 4.2.1 故障覆盖率 / 92
    - 4.2.2 测试生成中的故障仿真 / 94
    - 4.2.3 故障字典创建 / 95
  - 4.3 故障仿真技术 / 100
    - 4.3.1 串行故障仿真 / 102
    - 4.3.2 并行故障仿真 / 104
    - 4.3.3 并发故障仿真 / 107
    - 4.3.4 演绎故障仿真 / 109
    - 4.3.5 演绎故障仿真的比较 / 112
    - 4.3.6 关键路径追踪故障仿真 / 112
    - 4.3.7 微分故障仿真 / 115
  - 4.4 小结 / 115

**第 5 章 测试向量生成方法及算法 / 116**

- 5.1 测试生成基础知识 / 116
  - 5.1.1 布尔差分 / 116
  - 5.1.2 测试生成过程 / 118
  - 5.1.3 故障和测试 / 118
  - 5.1.4 术语和定义 / 119
- 5.2 可控性和可观察性 / 120
  - 5.2.1 可控性 / 120
  - 5.2.2 可观察性 / 120
  - 5.2.3 基于概率的可控性和可观察性 / 121
  - 5.2.4 SCOAP 的可控性和可观察性 / 126
  - 5.2.5 基于距离 / 130
- 5.3 随机测试生成 / 130
  - 5.3.1 限制随机测试数量 / 130
  - 5.3.2 组合电路随机测试生成 / 133
  - 5.3.3 时序电路的随机测试生成 / 139
- 5.4 小结 / 142

**第 6 章 确定性测试生成算法 / 143**

- 6.1 确定性测试生成方法 / 143
  - 6.1.1 双阶段测试生成 / 144
  - 6.1.2 面向故障的测试生成基本原理 / 144
  - 6.1.3 *D* 算法 / 149
  - 6.1.4 PODEM (面向路径的测试生成) / 156
  - 6.1.5 其他确定性面向故障的测试生成方法 / 161
  - 6.1.6 不依赖于故障的测试生成 / 162
- 6.2 时序电路测试生成 / 163
- 6.3 测试数据压缩 / 165
  - 6.3.1 测试压缩的形式 / 166
  - 6.3.2 测试兼容性 / 166

6.3.3 静态压缩 / 168

6.3.4 动态压缩 / 174

6.4 小结 / 174

**第 7 章 通过扫描法进行测试电路设计 / 175**

- 7.1 增加电路可测试性 / 175
  - 7.1.1 折中方案 / 175
  - 7.1.2 测试时序电路 / 176
  - 7.1.3 组合电路的可测试性 / 177
- 7.2 可测试性插入 / 177
  - 7.2.1 改善可观测性 / 177
  - 7.2.2 提高可控性 / 178
  - 7.2.3 共享可观测性引脚 / 180
  - 7.2.4 共享控制引脚 / 180
  - 7.2.5 降低选择输入 / 182
  - 7.2.6 同步控制和观测 / 182
- 7.3 全扫描可测试性设计技术 / 185
  - 7.3.1 全扫描插入 / 186
  - 7.3.2 触发器结构 / 187
  - 7.3.3 全扫描设计与测试 / 192
- 7.4 扫描结构 / 203
  - 7.4.1 全扫描设计 / 204
  - 7.4.2 映像寄存器可测试性设计 / 204
  - 7.4.3 局部扫描方法 / 206
  - 7.4.4 多扫描设计 / 209
  - 7.4.5 其他的扫描设计 / 210
- 7.5 RTL 扫描设计 / 211
  - 7.5.1 RTL 设计全扫描 / 211
  - 7.5.2 RTL 设计多链扫描 / 213
  - 7.5.3 RTL 扫描设计 / 215
- 7.6 小结 / 215

**第 8 章 标准 IEEE 测试访问方法 / 217**

- 8.1 边界扫描基础知识 / 217
- 8.2 边界扫描结构 / 218
  - 8.2.1 测试访问端口 / 218

- 8.2.2 BS-1149.1 寄存器 / 219
  - 8.2.3 TAP 控制器 / 223
  - 8.2.4 解码器单元 / 227
  - 8.2.5 选择器和其他单元 / 227
  - 8.3 边界扫描测试说明 / 227
  - 8.4 板级扫描链结构 / 233
    - 8.4.1 单一串行扫描链 / 234
    - 8.4.2 具有单一控制测试端口的多扫描链 / 234
    - 8.4.3 具有一个 TDI、TDO 但有多多个 TMS 的多扫描链 / 234
    - 8.4.4 多扫描链, 多 TAP / 235
  - 8.5 RTL 边界扫描 / 236
    - 8.5.1 为 CUT 插入边界扫描测试硬件 / 236
    - 8.5.2 两个模块的测试案例 / 239
    - 8.5.3 虚拟边界扫描测试机 / 239
  - 8.6 边界扫描描述语言 / 245
  - 8.7 小结 / 247
- 第 9 章 逻辑内建自测试 / 248**
- 9.1 内建自测试基本知识 / 248
    - 9.1.1 基于存储器的内建自测试 / 248
    - 9.1.2 内建自测试的有效性 / 250
    - 9.1.3 内建自测试的类型 / 250
    - 9.1.4 设计一个内建自测试 / 251
  - 9.2 测试向量生成 / 253
    - 9.2.1 测试向量产生器的集成 / 253
    - 9.2.2 穷举计数器 / 253
    - 9.2.3 环形计数器 / 254
    - 9.2.4 扭环计数器 / 255
    - 9.2.5 线性反馈移位寄存器 / 256
  - 9.3 输出响应分析 / 263
    - 9.3.1 输出响应分析器集成 / 263
    - 9.3.2 1 字符计数器 / 264
    - 9.3.3 跳变计数器 / 266
    - 9.3.4 奇偶校验 / 267
    - 9.3.5 串行 LFSR / 267
    - 9.3.6 并行特征信号分析 / 268
  - 9.4 内建自测试结构 / 270
    - 9.4.1 与内建自测试相关的术语 / 270
    - 9.4.2 集中式和独立式板级内建自测试结构 / 271
    - 9.4.3 内建评估和自检 / 272
    - 9.4.4 随机测试接口 / 273
    - 9.4.5 LSSD 片上自检 / 275
    - 9.4.6 使用 MISR 和 SRSG 自测试 / 276
    - 9.4.7 并发的内建自测试 / 278
    - 9.4.8 BILBO / 279
    - 9.4.9 提高测试覆盖率 / 280
  - 9.5 RTL 内建自测试设计 / 280
    - 9.5.1 被测电路设计、仿真和综合 / 281
    - 9.5.2 RTS 内建自测试插入 / 281
    - 9.5.3 配置 RTS 内建自测试 / 286
    - 9.5.4 内建自测试的合并配置 / 289
    - 9.5.5 STUMPS 设计 / 289
    - 9.5.6 RTS 和 STUMPS 的结果 / 292
  - 9.6 小结 / 292
- 第 10 章 测试压缩 / 293**
- 10.1 测试数据压缩 / 293
  - 10.2 压缩方法 / 295
    - 10.2.1 基于代码的方案 / 295
    - 10.2.2 基于扫描的方案 / 303
  - 10.3 解压缩方法 / 309
    - 10.3.1 解压缩的硬件结构 / 309
    - 10.3.2 周期性扫描链 / 311
    - 10.3.3 基于代码的解压缩 / 312
    - 10.3.4 基于扫描的解压缩 / 317
  - 10.4 小结 / 317

**第 11 章 通过 MBIST 测试存储器 / 318**

- 11.1 存储器测试 / 318
- 11.2 存储器结构 / 319
- 11.3 存储器故障模型 / 320
  - 11.3.1 固定故障 / 320
  - 11.3.2 转换故障 / 320
  - 11.3.3 耦合故障 / 320
  - 11.3.4 桥接和状态耦合故障 / 321
- 11.4 功能测试方法 / 321
  - 11.4.1 March 测试算法 / 321
  - 11.4.2 March-C 算法 / 322
  - 11.4.3 MATS+ 算法 / 322
  - 11.4.4 其他的 March 测试 / 322
- 11.5 MBIST 方法 / 323
  - 11.5.1 简单的 March MBIST / 323

11.5.2 March-C MBIST 计数 - 排序器 / 328

11.5.3 干扰 MBIST / 331

11.6 小结 / 332

**附录 A 在协议感知自动测试设备上使用 HDL / 333**

**附录 B PLI 测试应用的门级组件 / 336**

**附录 C 编程语言接口测试工具 / 338**

**附录 D IEEE 1149.1 标准边界扫描的 Verilog 描述 / 343**

**附录 E 边界扫描 IEEE 1149.1 标准虚拟测试机 / 349**

**附录 F 由 RTL 综合生成的门级网表 (NetlistGen) / 359**

**参考书目 / 362**



# 数字电路测试的基础知识和 HDL 的作用

本章介绍了一些与数字系统测试相关的基础知识。本章重点介绍测试中各种 HDL 的使用，介绍了基于 HDL 的设计流程，讨论了如何将各种具体的测试应用插入到该流程中，并给出了不同场合下如何使用 HDL 辅助数字系统测试。

本章试图回答一些与数字系统测试相关的重要问题。首先回答的问题是：在数字系统中到底测试什么，以及为什么要对其进行测试。答案是：测试是为了发现成品芯片或器件的物理缺陷（详情见后续章节）。其他有待回答的问题包括：采用的测试方法；提高芯片或成品器件可测试性的方法；HDL 是如何辅助测试流程的；以及在数字系统测试中我们所面临的一些局限。

学完本章之后，就不难理解本书为何用了三章来讲述测试方法，又用了另外三章来讲述如何使电路具备可测试性，同时，在这些章节中，我们也多次重申要减少测试数据量和测试时间。此外，通过在本章中对 HDL 基础知识进行讨论，你就能理解在本书其他章节中 HDL 的各种不同使用方式。

接下来的部分讨论基于 HDL 的设计以及把测试插入芯片设计流程的位置。紧接着的四小节讨论了测试重点、测试方法以及处理此类测试重点的可测试性方法。1.3 节讨论了 HDL 在设计和测试流程中所能发挥的作用，本章所述的 HDL 知识也贯穿于后续章节。1.4 节讨论了在成形测试及可测试性技术中起着重要作用的测试设备。

## 1.1 设计及测试

数字系统的制造流程，是以设计师采用高级设计语言详细描述其设计开始，以制造并装运各部件交付客户为终点。整个流程涉及了仿真、综合、测试等多个阶段，这些都会在本书中加以说明。

### 1.1.1 RTL 设计流程

在寄存器传输级（RTL）设计流程中，设计师首先会用 Verilog 等 RTL 语言实现其芯片设计规格。如图 1-1 所示，该描述图采用了 always 语句等 Verilog 高级语法结构，通过使用同一