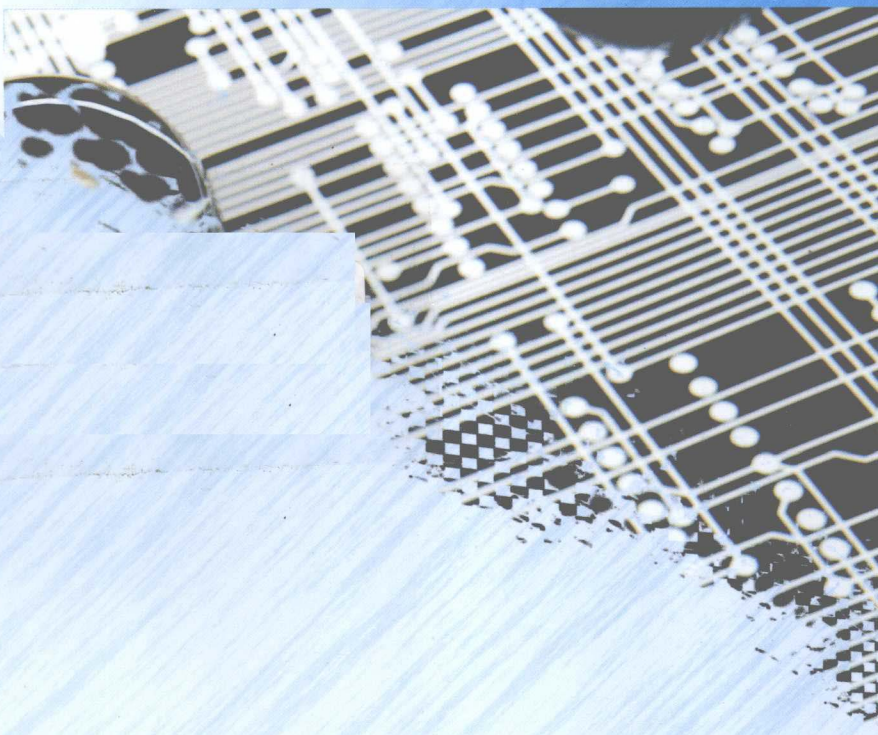
李晓维　吕 涛　李华伟　李光辉 著

# 数字集成电路设计验证

## ——量化评估、激励生成、形式化验证



科学出版社
www.sciencep.com

# 数字集成电路设计验证

## ——量化评估、激励生成、形式化验证

李晓维　吕　涛　李华伟　李光辉　著

科学出版社

北　京

## 内 容 简 介

本书内容涉及数字集成电路设计验证的三个主要方面：量化评估、激励生成和形式化验证。主要包括寄存器传输级（RTL）电路建模、基于可观测性的覆盖率评估方法、设计错误模型；基于故障模型的激励生成、基于 RTL 行为模型的激励生成、覆盖率驱动的激励生成；基于可满足性的等价性检验、包含黑盒电路的形式化验证，以及不可满足问题。

全书图文并茂，阐述了作者及其科研团队自主创新的研究成果和结论，对致力于数字集成电路设计验证方法研究的科研人员（尤其是在读研究生），具有较大的学术参考价值，也可用作集成电路专业的高等院校教师、研究生和高年级本科生的教学参考书。

**图书在版编目(CIP)数据**

数字集成电路设计验证：量化评估、激励生成、形式化验证/李晓维等著. —北京：科学出版社，2010

ISBN 978-7-03-027609-4

Ⅰ.①数… Ⅱ.①李… Ⅲ.①数字集成电路-电路设计-验证 Ⅳ.①TN431.2

中国版本图书馆 CIP 数据核字(2010)第 089305 号

责任编辑：刘宝莉/责任校对：刘小梅

责任印制：赵 博/封面设计：鑫联必升

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

漷海印刷有限责任公司 印刷

科学出版社发行 各地新华书店经销

*

2010 年 5 月第 一 版 开本：B5（720×1000）

2010 年 5 月第一次印刷 印张：26 3/4

印数：1—3 000 字数：518 000

**定价：58.00 元**

（如有印装质量问题，我社负责调换）

# FOREWORD

Verifying that a hardware design complies with its functional specifications continues to be a major challenge. It has been reported that up to 40% of the human resources and 70% of the computing resources in a typical high-end microprocessor design project are devoted to the verification task. While this is understandable (releasing buggy designs can be quite costly economically and otherwise), it still leaves one to wonder why such a mechanical task should consume so much of a design team's time and talent, resources that are better deployed to the more creative aspects of design. What's even more disconcerting is that the current approaches to verification (simulation, emulation, and formal methods) can hardly keep up with the exponential increase in design complexity-evident from the increasing numbers of bug-escapes through fabrication and consequently expensive re-spins for each new generation of design.

The Joint Research Laboratory for Advanced Test Technology at Institute of Computing Technology of Chinese Academy of Sciences has been very active in researching verification technologies and has produced a broad range of impactful results on coverage metrics, vector generation, equivalence checking, and model checking. While individual results were published as PhD theses as well as conference and journal papers, this book provides a self-contained, unified, and in-depth coverage of these timely topics in a coherent fashion and thus offers significant additional values beyond the collection of theses and papers. The coverage of each topic includes motivations and problem definitions, a survey of prior arts, detailed descriptions of new advances, and demonstrations of experimental results and comparisons. Therefore, this book will allow the readers to gain better understanding of both the fundamentals as well as more advanced development for these important subjects.

The total state space for complete verification is well beyond what can be practically checked. Thus some modeling and analysis techniques are necessary for estimating the degree of completeness of the verification tasks as well as for guiding the selection of additional test cases and/or design modifications for enhancing verifiability. Many design and verification teams today use code-, functional-, and/or assertion-coverage to estimate verification coverage while it is universally acknowledged that each type of models or metrics often has only limited

and very specific application and thus effectiveness of these metrics remain design- and flow-dependent. In Part I of the technical content of the book, including Chapters 2 to 5, gives a thorough treatment of this subject and reports some exciting new research results to address the limitations of existing models and metrics.

Automatic generation of high-quality vectors remain a central research focus for verification of debugging. Part II of the technical content ,including Chapters 6 to 9, describes error-model-driven and coverage-driven vector generation. Both deterministic and guided-random approaches are discussed.

The promise of formal verification has been hailed, by some, as a panacea. Alas, formal methods have failed to convince the skeptics and have somewhat fallen short of that promise. To be sure, some aspects of design verification are now routinely performed with tools that employ formal reasoning; for example, establishing the equivalence between two implementations of the same design is now regularly done using formal equivalence checking tools which have more or less replaced simulators as the preferred method. Property verification via model checking, while has not caught on to the same extent, is gaining increasingly popularity. Part III of the technical content of the book, including Chapters 10 to 14, addresses the scalability and computational efficiency of various formal methods. It starts with an introduction to formal verification approaches, followed by a thorough coverage of incremental equivalence checking, optimization techniques to the Boolean satisfiability problem, and a number of core issues for model checking.

The fast and continuing evolution of verification technology and the enormous growth in the complexity and sophistication of the coverage and verification tools has made it such that very few people can really master all fronts of this subject. New problems, new algorithms, and new methodologies and tools continue to emerge. As a result, it is becoming difficult even for experts to follow and comprehend the progress on a continuing basis. This book organizes a huge body of complex materials into a logical and easy-to-understand fashion and thus will help graduate students, engineers, and researchers to get a better grasp of recent advances and gains better insight into this fast evolving field.

Kwang-Ting (Tim) Cheng (郑光廷)
Univ. of California, Santa Barbara
December 4, 2009

# 前　言

实现所期望的功能特性是集成电路设计需要满足的最基本要求，功能验证是集成电路设计验证的基础技术，是集成电路设计的关键技术之一，是确保集成电路功能正确性的主要技术手段。

本书是全面论述数字集成电路设计验证方法的学术著作，汇集了自2000年以来中国科学院计算研究所（以下简称中科院计算所）在数字集成电路设计验证方法学研究中取得的自主创新的重要研究成果和结论。内容涉及数字集成电路设计验证的三个重要方面：量化评估、激励生成和形式化验证。

全书共15章，其中技术内容可分为三大部分。第一部分（第2～5章）量化评估，从可观测性信息和发现设计错误的能力两个角度，论述数字集成电路设计验证的量化评估方法。第二部分（第6～9章）激励生成，针对寄存器传输级的激励生成问题，从故障模型和覆盖率导向两个角度，论述确定性和非确定性的激励生成方法。第三部分（第10～14章）形式化验证，从提高处理速度的角度，论述形式化验证中的等价性检验方法和模型检验方法。

本书的主要技术内容汇集了李晓维研究员从2001年以来指导的博士生（李光辉、吕涛、鲁巍、邵明等）和硕士生（刘领一、赵阳、王天成等）的学位论文工作的部分成果；也包括李华伟和尹志刚的博士学位论文的部分成果。这些研究成果部分已经在本领域相关学术刊物和学术会议上发表。本书由李晓维研究员主持撰写，吕涛博士参与了第3～5章内容的整理；李华伟研究员参与了第2、6～9章内容的整理；李光辉教授参与了第10～14章内容的整理。清华大学计算机系边计年教授审阅了全部书稿，美国UCSB计算机系主任郑光廷教授撰写了序言。在此表示衷心的感谢。

本书汇集的部分科研成果是在国家重点基础研究计划（973）课题“高性能处理芯片的设计验证与测试”（2005CB321605）、国家自然科学基金重点项目“数字VLSI电路测试技术研究”（60633060）和“从行为级到版图级的设计验证与测试生成”（90207002）等资助下完成的。研究过程中得到了中科院计算所李国杰院士、闵应骅研究员、胡伟武研究员、李忠诚研究员等领导和同事的关心和支持，得到了清华大学边计年教授、复旦大学唐璞山教授、浙江大学严晓浪教授、西安邮电大学韩俊刚教授等同行的支持和帮助，在此表示衷心的感谢。

由于作者水平和经验有限，书中难免存在不足之处，恳请读者批评指正。

# 目　录

# 第1章 绪 论

验证是一个使用非常广泛的词汇。在电子设计自动化(electronic design automation,EDA)领域,所谓设计验证(design verification),指的是判定一个设计的实现(implementation)是否满足其规范(specification)。一个电子器件通常具有很多方面的特性,包括功能、性能、功耗等。相应的,设计验证也包括了很多方面的技术,如功能验证(functional verification)技术、静态定时分析技术、功耗分析技术等。因此,验证这个词在不同的上下文中有着不同的含义。本书着眼于功能验证技术。实现所期望的功能特性是一个电路设计需要满足的最基本要求。如果没有特别声明的话,后面提到的设计验证均指对功能的验证。

众所周知,在产品的设计流程中,设计错误(design error,通俗来讲即设计中的 bug)发现得越晚则带来的损失越大。著名的 Intel 公司的 Pentium 处理器的浮点除法错误,导致了约 4 亿美元的损失[1]。由此可见,在芯片(chip)设计流程中实施充分的有效的设计验证工作,其重要性非同寻常。

随着集成电路工艺的不断细化,设计验证在集成电路设计流程中所耗费的开销越来越大。2007 年度的《国际半导体技术发展报告》(*The International Technology Roadmap for Semiconductors*)指出,在当前的工程项目中,验证工程师的人数超过了设计工程师,对于复杂的设计更是达到了 2∶1 的比例[2]。

工业需求推动着验证技术的科学研究工作不断升温,受到越来越多的学者的关注。2007 年度的图灵奖就授予了提出模型检验技术的 Edmund M. Clarke 等人,这充分表现了科学界对于设计验证技术的关注。电路设计的规模和复杂性随着半导体制造工艺的细化呈上升趋势,如果没有重大突破的话,验证问题将是半导体工业发展的重大障碍。

## 1.1 设计验证简介

在常见的数字集成电路设计流程中,不同的设计环节对应着不同抽象层次的设计。图 1.1 是常见的不同抽象层次的设计及验证。依照抽象程度从高到低的顺序,分别是功能规范、算法级/微体系结构级设计、寄存器传输级(register-transfer level,RTL)设计、门级(gate level)设计和物理级设计。相应的,对某个设计的验证任务,常常被转化为该设计与相邻的具有较高抽象层次的设计之间是否相符的问题。在门级和物理级这样的较低抽象层次上,设计验证技术已经相

对成熟，尤其是形式化的等价性检验技术，已经被业界广泛采用。而在 RTL 以上的抽象层次上，设计验证技术虽然也取得了不断的发展和进步，但是尚没有完善的解决方案。这方面的技术包括基于模拟的验证技术，以及形式化验证中的模型检验和定理证明。

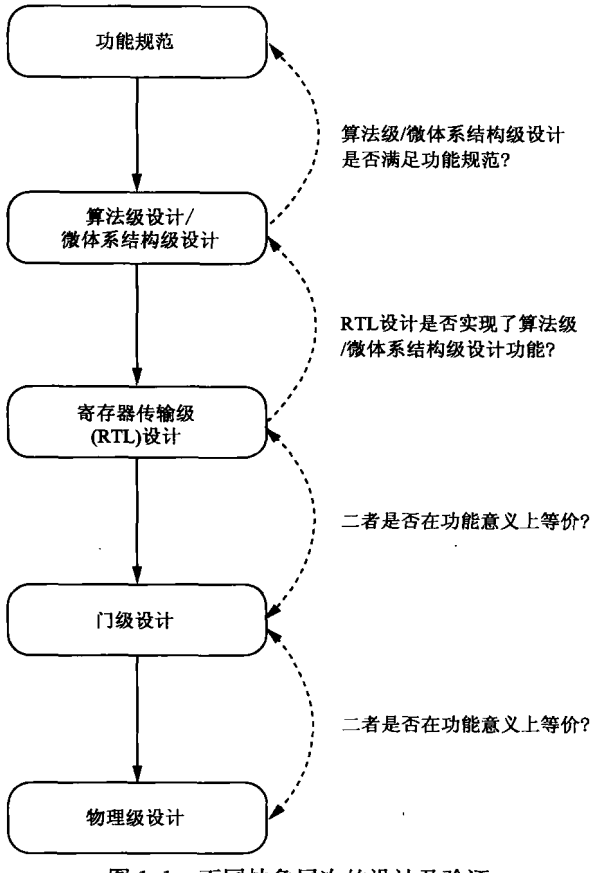



图 1.1 不同抽象层次的设计及验证

模拟是业界常用的，也是最主要的验证技术。开发模拟验证环境所用的语言经历了几次更新换代，从直接采用硬件描述语言(hardware description language，HDL)，到采用面向对象的语言(如 C++)，再到专门的验证语言(如 SystemVerilog 语言和 e 语言)。专门的验证语言使得验证工程师可以方便地开发出功能强大的验证环境。模拟验证中的典型验证环境如图 1.2 中虚线内所示。图 1.2 中，矩形结点表示验证环境的组成模块，虚线外部的结点表示与模拟验证过程有关的文档，连线表示信息或数据的流向关系。

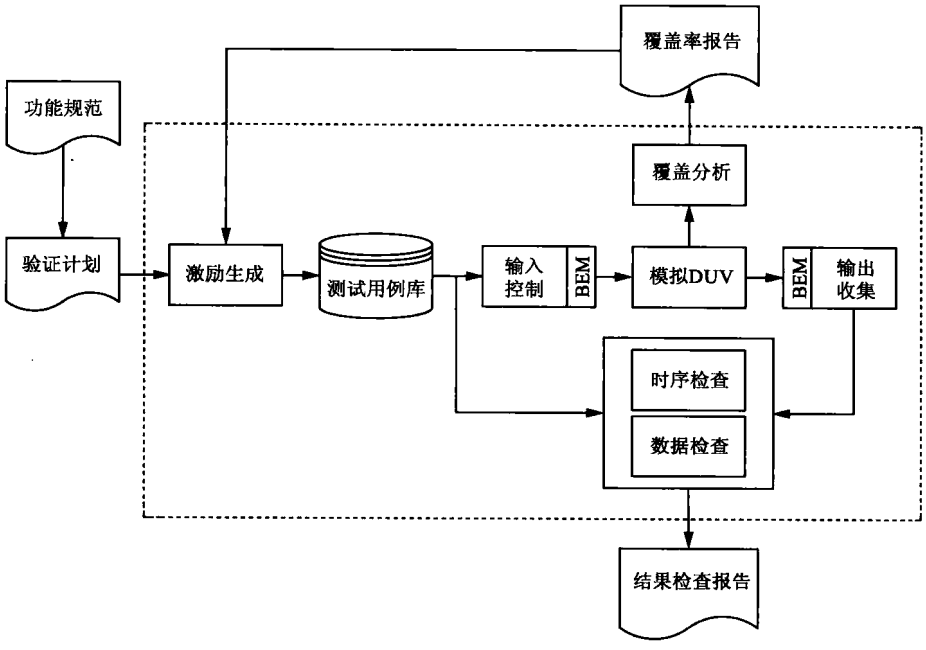



图 1.2 典型的模拟验证环境

在模拟验证过程中，验证工程师首先需要理解功能规范并制定出相应的验证计划，然后依照验证计划开发测试用例(test case)。除了定制一些测试用例之外，经常采用的技术还有基于约束的随机激励生成(constraint-based random vector generation)。为了提高验证环境的开发效率和验证环境中模块的可重用性，验证语言通常采用类似面向对象语言的方式来描述测试用例。因此在向被验证设计(design under verification，DUV)，尤其是 RTL 或门级的 DUV 施加测试用例时，需要将较抽象的测试用例映射成二进制向量①(vector)，必要时还需要通过总线功能模型(bus functional model，BFM)[3]按照 DUV 所需要的时序关系来施加二进制向量，这就是图 1.2 中输入控制部分需要实现的功能。输出收集部分的功能则是输入控制部分功能的逆过程。对于模拟过程中收集到的数据，一方面需要进行结果检查，包括时序检查和数据检查，形成结果检查报告以反映模拟验证过程中是否发现了设计错误；另一方面需要进行覆盖率分析，形成覆盖率报告以评估模拟验证的充分程度。对于设计验证环境中各部分的详细介绍，可见文献[4]。

模拟验证技术需要构造各种测试用例，对于大型复杂的设计，模拟会消耗大量的时间。然而，由于市场的压力，又必须尽可能地缩短设计周期来加快产品面市。

①在后面的章节中，当不强调功能场景时，通常不讲“测试用例”，而直接讲“向量”、“输入向量”或者“激励”。