

电子系统测试原理

Principles of Testing Electronic Systems

(美) Samiha Mourad 著
Yervant Zorian

张威 王仲 等译 宫云战 审校

PRINCIPLES
OF
TESTING
ELECTRONIC
SYSTEMS

SAMIHA MOURAD
YERVANT ZORIAN

TN06
电子与电气工程丛书

TN06

11

2007

电子系统测试原理

Principles of Testing Electronic Systems

(美) Samiha Mourad 著
Yervant Zorian

张威 王仲 等译 宫云战 审校



机械工业出版社
China Machine Press

本书全面阐述了电子系统测试原理, 共分为五个部分, 第I部分介绍本书的目的、电子产品的缺陷种类、VLSI设计表示与设计流程; 第II部分介绍故障模拟、测试码产生及电流测试方法; 第III部分讨论易测试性设计问题, 分别介绍专用技术、路径扫描设计、边界扫描测试和内建自测试技术; 第IV部分介绍特殊结构的测试, 包括存储器、FPGA和微处理器的测试; 第V部分涉及当前电子系统测试领域中的前沿问题, 包括如何实现易测试性结构和进行SOC测试。

本书可作为电子信息类专业研究生教材和IC测试工程技术人员的参考书。

Samiha Mourad, Yervant Zorian: Principles of Testing Electronic Systems (ISBN:0-471-31931-7)

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

Copyright © 2000 by John Wiley & Sons, Inc.

All rights reserved.

本书中文简体字版由约翰-威利父子公司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。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2003-7217

图书在版编目(CIP)数据

电子系统测试原理 / (美) 莫瑞达 (Mourad, s.), (美) 佐瑞安 (Zorian, Y.) 著; 张威、王仲译. - 北京: 机械工业出版社, 2007.1

(电子与电气工程丛书)

书名原文: Principles of Testing Electronic Systems

ISBN 7-111-19808-5

I. 电… II. ①莫… ②佐… ③张… ④王… III. 电子系统-测试技术 IV. TN06

中国版本图书馆CIP数据核字(2006)第098218号

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: 傅志红

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007年1月第1版第1次印刷

184mm×260mm·19.25印张

定价: 39.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

译者序

综观IC测试技术近五十年的发展史，其发展历程是和IC技术的发展相辅相成的。首先IC在规模上的增加，从几个门电路到上千万乃至上亿的门电路数量，促使IC测试技术必须探索更快的测试生成算法，从20世纪60年代以来，许多形式化、智能化的测试生成算法和故障仿真算法被相继提出。其次，IC技术的快速发展，绝不仅仅表现在门电路数量上的增加，电路的复杂性也在大幅度地增加，智能芯片、SOC、RAM、FPGA等特殊电路，给测试带来了极大的挑战，新的测试技术必须应付这种挑战；同时电路的复杂性也导致了其制造工艺上的革命性变革，新的故障模式与新的测试方法也是必须要研究的问题。再次，测试技术绝不是只能完全被动的适应于制造，从系统科学的角度来看，复杂系统的设计必须要考虑测试性设计问题，这可以降低VLSI设计成本。最后，对自动测试系统而言，测试者必须要了解IC的设计过程与设计表示。《电子系统测试原理》的英文版是美国的Samiha Mourad和Yervant Zorian先生于2000年最新出版的关于IC测试技术的专著，内容丰富，基本涵盖了当今IC测试技术领域的全部内容。而在此之前的相关的资料中，还没有这么全面的一本著作。

全书分为五个主要部分。第一部分是测试技术综述，阐述了测试技术的一般方法、与测试相关的问题和故障模式；对自动测试而言，设计过程与数据是必要的，因此本部分也对VLSI的设计表示与设计过程进行了综述。第二部分是测试流，论述了仿真的一般方法和故障仿真方法，基于固定型故障模型的测试码生成的一般方法，以及基于电流模型的测试方法。第三部分论述了测试性设计技术，包括测试性设计技术的一般概念和基本方法、扫描路径设计技术、边界扫描设计技术和内部自测试技术。第四部分论述了特殊结构电路的测试技术，包括内存测试技术、FPGA与微处理器的测试技术。第五部分是前沿技术论题，主要论述了测试性综合技术的一般方法和SOC的测试技术。

本书结构清晰、内容全面、编写严谨，是电子系统测试方面的经典著作。

本书由张威、王仲、卢庆龄、高传平、金大海、杨洪路、毕学军等翻译，由宫云战教授审校。

在翻译过程中得到了中国科学院计算技术研究所的闵应骅研究员、李晓维研究员和李华伟研究员的热情指导和帮助，在此深表感谢。由于译者水平所限，不当之处在所难免，欢迎读者批评指正。

译者

2006年10月

前 言

过去十年来,由于工艺器件特征尺寸的减小、芯片容量的扩大,以及设计易测试电路的复杂性,电子系统测试面临越来越大的挑战。随着电子系统的发展,自动设计工具也得到了长足的进展。实际上,目前设计过程的每一个阶段(包括高级综合阶段)都实现了自动化。早期只在门级电路考虑易测试性,但当今电路的复杂性和片上系统(SOC)的出现,使得在设计过程之初就必须考虑测试问题。所有这些因素给设计和测试工程师带来了诸多的挑战,这些挑战包括(但不局限于):研究更精确的故障模型,在高层设计表示上检查易测试性,以及在综合之前或综合过程中嵌入更有效的测试结构。VLSI测试的另一个主要挑战是IC要经受大量测试数据的考验。

技术快速变化和发展的同时也带来了一系列设计和测试问题,有效解决这些问题显得越来越迫切,因此需要大量高素质的工程师来承担这一任务。许多工作在一线的工程师主要专心于测试和测试设计工作,很少有机会系统地研究这些问题。他们主要在工作中学习知识,虽然通过实践获取知识是一种有效的方式,但在工作之前掌握一些测试理论和基本知识,效果可能会更好。本书就是为这些具备一定测试基础的工程师而写的,其主要内容包括:

- 设计过程中的哪些阶段需要考虑测试;
- 详细讨论了路径扫描和扫描链寄存器顺序;
- 随机逻辑和存储器的自测试方法;
- 基于RAM的FPGA的测试方法;
- 片上系统的测试。

要具备扎实的测试基础,必须理解五个基本规则:

- 电路级的物理缺陷及其表现;
- 检测故障的测试码生成;
- 测试和设计周期的关系;
- 测试实践设计;
- 制造芯片的测试。

本书详细讨论了前四个规则,采用的方法并不是纯理论推导,而是站在理论的高度通过现象来了解主题。例如,可以采用纯数学方法来阐述测试码产生的D算法,但采用流程图来描述算法足以表达清楚,并且更适于在软件工具中实现。这并不排斥深入的理论研究,如果读者要进一步研究基础理论,本书的参考文献中给出了一系列的专业测试著作、期刊杂志和Web站点。

本书分为五个部分。第I部分共四章,是本书的综述部分。为了获得可靠的电子产品,该部分一开始就涉及了设计和测试问题。第1章介绍了本书的目的,以及设计验证和测试的区别。设计验证的目的是检查设计是否满足需求说明,而测试的目的是验证硅片上的设计实现是没错的。此外,该章还简要介绍了测试最终产品所需要做的准备工作及采用的步骤和工具,即

测试码产生和设计,以提高产品的易测试性。

在详细讨论第1章所介绍的主题之前,了解产品失效的原因是非常重要的。第2章全面介绍了电子产品可能遇到的缺陷,并解释了把这些缺陷与故障映射到设计结构和行为级上的必要性。大多数电子产品的失效是由生产过程中的缺陷引起的,但有些失效也可能由产品操作中的干扰引起。由于绝大多数VLSI系统都使用了CMOS,因此重点介绍MOS技术。

在产品设计初期就应该考虑测试问题。在产品生命周期的不同阶段,了解如何表示设计是非常重要的。为此,第3章介绍了设计分类方法,这些方法可以用来解释第4章介绍的设计周期的各个活动。自动设计过程是依据各种算法的综合性活动,算法通常用图表来表示。掌握这部分知识有两个作用。首先,有利于了解本书的后续内容。其次,有助于建立设计和测试是不可分的概念。

第II部分包含三章内容。尽管目前在CAD工具中更多地使用形式验证技术,但模拟仍然是设计验证的主要手段,它在产品设计和测试中起着举足轻重的作用。此外,故障模拟有助于评估产品测试过程中产生的测试码的质量。第5章介绍了故障模拟如何促进了当前非常密集的大规模集成电路的发展。

第6章介绍了什么是确定性测试码生成。测试码生成一般依赖于检测电路输出的电压信号,但现在除了检测电压之外,电流测量也被使用,这就是所谓的 I_{DDQ} 测试,它只适用于CMOS电路,而当前大多数电子电路都是CMOS电路。第7章展示了当今流行的测试方法,它能有效检测电压测试所不能检测的缺陷。这种新的缺陷检测范例正变得越来越重要,并且也被用于传统的故障检测。而且,电流测试有利于缺陷诊断。

第III部分基于前七章介绍的知识讨论了易测试性设计问题。这一部分包含四章内容。第8章概述了DFT的通用方法。特殊(Ad hoc)技术虽然非常简单,但功能非常强大。例如,在测试产品时采用分而治之的方法可以发现细微的差异。随后的三章介绍了结构化技术。第9章展示了一种减少同步时序电路复杂性的方法,即路径扫描设计方法,它使用非常广泛,并且正成为电子电路的标准功能,它还可以用于异步电路中。第10章介绍了边界扫描设计,该技术最初用于印制电路板,但现在也非常适用于集成电路。这种DFT技术遵循IEEE标准1149.1,并且可以用于调试和分析。第11章是这一部分的最后一章,它介绍了内建自测试(BIST)技术,能有效应用于随机逻辑,还可以用于内存测试、微处理器和FPGA。该章首先解释了其基本原理,然后描述了它与扫描设计在随机逻辑中的应用问题。

第IV部分包含两章内容。介绍了测试如何执行,以及通用RAM、FPGA和微处理器的测试。该部分介绍的三种结构都利用了功能故障模型,因为使用结构故障模型使测试难以跟踪。基于功能模型的测试还可以检测诸如译码逻辑和附属寄存器等电路的失效问题。

第12章介绍了RAM测试。RAM的结构虽然很普通,但是电路非常密集,并可能引起一系列测试问题。当RAM嵌入逻辑芯片或SOC时,这些问题更为突出。第13章介绍了另外两种通用电路。FPGA有很多种类,这里只介绍基于RAM的FPGA。它们也拥有通用数组结构,但没有RAM的电路密集。该章研究了它们的通用结构,并使其具有C易测试性。微处理器在某种程度上是使用最多的器件,但它们的易测试性并没有被充分研究;用于微处理器测试的模型几乎没有。大多数嵌入式易测试性结构,如路径扫描和BIST,可以广泛用于微处理器测试。

第V部分是全书内容的总结。正如我们所倡导的,在设计周期中应该考虑测试问题,并且在设计时考虑易测试性结构。有两种实现易测试性的方法,一种是分两步走,首先完成设计,

然后添加易测试性结构，但这通常会降低电路性能。另一种方法是一步到位，即在设计周期开始的时候就考虑易测试性问题，这样设计的电路在面积、性能和易测试性方面都能得到优化。第14章介绍了如何在综合过程中添加约束条件以达到一步到位的方法。第15章综合运用前面介绍的知识讨论了遇到的新问题，这些问题是在同一个IC上嵌入大量预设计块引起的。测试数据的数量增加了，但效率仍然较低。SOC测试目前仍然是测试领域里的一个挑战。

本书综合了其中一位作者的长期教学经验和其他作者的丰富企业经验，力图做到既适合教学又适合测试实践参考。

致谢

本书的完成受到几个朋友、同事和学生的启发与帮助，在此对他们表示衷心的感谢，尤其是加拿大多伦多大学的Sunil Das，加利福尼亚州门洛帕克SUN公司的Yacoub M.EIZiq，加利福尼亚州圣克鲁斯加利福尼亚大学的F.Joel Ferguson，加利福尼亚州圣克拉拉Cabeltron公司的Suresh Gopalakrishnan，加利福尼亚州桑尼维尔Marvell公司Daryl Hoot，加利福尼亚州桑尼维尔Toshiba公司London Jin，加利福尼亚州圣克拉拉Transmeta公司的Samy Makar，北卡罗来纳州夏洛特北卡罗来纳大学的Rafic Makki，日本东京大学的Takashi Nanya，科罗拉多州柯林斯堡Hewlett Packard公司的Kenneth P.Parker，加利福尼亚州圣克拉拉大学的Prachi Sathe，以及荷兰Delft理工大学的A.J.van de Goor。

此外，还要感谢George Telecki和Cassie Craig，感谢他们有价值的建议，以及在本书手稿几次推迟交稿时给予我们的耐心。感谢Martha Giannini、Mike Daly和Amit Hakoo，没有他们的帮助，手稿是不可能完成的。圣克拉拉大学的许多学生首先阅读了本书的草稿，感谢他们有益的评论和建议。最后，感谢所有对本书提供帮助的人们。

Samiha Mourad

(加利福尼亚州圣克拉拉大学)

Yervant Zorian

(Logic Vision公司)

目 录

译者序
前言

第I部分 设计与测试

第1章 测试综述	1
1.1 可靠性与测试	1
1.2 设计过程	1
1.3 验证	4
1.3.1 功能模拟	4
1.3.2 时间模拟	4
1.4 测试	4
1.5 故障及其检测	6
1.6 测试码生成	7
1.7 故障覆盖率	8
1.8 测试类型	8
1.8.1 穷举测试	8
1.8.2 伪穷举测试	8
1.8.3 伪随机测试	9
1.8.4 确定性测试	9
1.9 测试应用	9
1.9.1 在线测试与离线测试	10
1.9.2 自动测试仪器	10
1.9.3 片上测试与片外测试	11
1.10 易测试性设计	11
1.10.1 可控性	12
1.10.2 可观察性	12
1.11 测试经济	12
1.11.1 收益和缺陷级	13
1.11.2 故障覆盖率和缺陷级别	13
1.12 进一步研究	14
参考文献	15
习题	16
第2章 缺陷、失效和故障	18
2.1 简介	18

2.2 物理缺陷	19
2.2.1 材料过多和缺失	20
2.2.2 氧化物断裂	20
2.2.3 电迁移	20
2.3 故障模式	21
2.3.1 开路	21
2.3.2 短路	21
2.4 故障	22
2.5 固定型故障	22
2.5.1 单固定型故障	22
2.5.2 多固定故障	24
2.6 故障列表	24
2.6.1 等价关系	24
2.6.2 支配关系	25
2.6.3 故障精简	25
2.7 桥接故障	26
2.8 短路和开路故障	28
2.8.1 NMOS电路	29
2.8.2 CMOS电路	29
2.9 时延故障	32
2.10 暂时失效	33
2.10.1 瞬时故障	33
2.10.2 间歇故障	34
2.11 噪声失效	34
参考文献	35
习题	38
第3章 设计表示	39
3.1 抽象级	39
3.2 数学方程	41
3.2.1 开关函数	41
3.2.2 布尔差分	42
3.2.3 有限状态机	43
3.2.4 晶体管级表示	43
3.3 列表格式	45

3.3.1 真值表	45	5.4.1 编译模拟	79
3.3.2 状态表	45	5.4.2 事件驱动模拟	80
3.4 图形表示	46	5.5 时间模型	81
3.5 图	47	5.5.1 静态时间分析	83
3.6 二叉判断图	49	5.5.2 混合级模拟	83
3.7 网表	51	5.6 故障模拟	83
3.8 硬件描述语言	52	5.6.1 并行故障模拟	84
3.8.1 Verilog语言	52	5.6.2 演绎故障模拟	85
3.8.2 VHDL语言	54	5.6.3 并发故障模拟	87
参考文献	54	5.7 故障模拟结果	89
习题	55	5.7.1 故障覆盖率	89
第4章 VLSI设计流程	57	5.7.2 故障字典	90
4.1 简介	57	参考文献	91
4.2 CAD工具	57	习题	91
4.3 算法	58	第6章 自动测试码生成	93
4.4 综合	59	6.1 简介	93
4.4.1 行为综合	62	6.2 术语和符号	93
4.4.2 逻辑综合	62	6.2.1 基本操作	93
4.5 设计方法	63	6.2.2 逻辑和集合操作	94
4.6 半定制设计	64	6.2.3 故障列表	95
4.6.1 标准单元设计	64	6.3 D算法	96
4.6.2 掩模可编程门阵列	65	6.3.1 内部节点情况	97
4.6.3 可编程设备	65	6.3.2 原始输入情况	98
4.7 物理设计	67	6.3.3 原始输出情况	99
4.7.1 平面布局	67	6.3.4 选择策略	99
4.7.2 布局	69	6.4 临界路径	99
4.7.3 布线	70	6.5 回溯和扇出重汇聚	101
4.7.4 反标	72	6.6 PODEM	101
参考文献	73	6.7 其他算法	105
习题	74	6.7.1 FAN算法	105
		6.7.2 SOCRATES	105
		6.8 时序电路测试	105
		6.8.1 功能测试	106
		6.8.2 确定性测试码生成	109
		参考文献	112
		习题	114
		第7章 电流测试	116
		7.1 简介	116
		7.2 基本概念	117

第II部分 测试流程

第5章 测试中模拟的角色	77
5.1 简介	77
5.2 大型设计的模拟	78
5.2.1 测试平台	78
5.2.2 基于设计周期的模拟	79
5.3 逻辑模拟	79
5.4 模拟方法	79

7.3 无故障电流	119	9.4 测试码应用	155
7.3.1 转换与静止电流	119	9.4.1 测试触发器	156
7.3.2 转换时延	120	9.4.2 测试电路的组合部分	156
7.4 电流感应技术	121	9.5 路径扫描设计的例子	156
7.4.1 片外测量	121	9.6 存储设备	158
7.4.2 片上测量	122	9.6.1 两端口触发器	158
7.5 故障检测	123	9.6.2 时钟门锁	159
7.5.1 泄漏故障	124	9.7 扫描结构	160
7.5.2 桥接故障	125	9.7.1 级敏扫描设计	160
7.5.3 固定开路故障	125	9.7.2 扫描集结构	161
7.5.4 时延故障	126	9.8 多级扫描链	162
7.6 测试码生成	127	9.9 路径扫描设计的代价	162
7.6.1 基于开关级模型	127	9.9.1 额外区域与引脚	162
7.6.2 基于泄露模型故障	128	9.9.2 性能	163
7.7 深亚微级技术的影响	128	9.9.3 测试时间	163
参考文献	130	9.9.4 热消耗	163
习题	132	9.10 部分扫描测试	163
 第III部分 易测试性设计		9.10.1 定义	165
第8章 专用技术	133	9.10.2 选择扫描触发器	165
8.1 简介	133	9.10.3 测试应用	166
8.2 DFT的内容	133	9.11 调整扫描链上的触发器	166
8.2.1 测试码产生及应用	133	9.11.1 最优化测试应用	166
8.2.2 当前大规模集成电路特性	134	9.11.2 最优化连接线	168
8.3 易测试性分析	135	参考文献	169
8.4 初始化及测试点	138	习题	170
8.4.1 初始化	138	第10章 边界扫描测试	172
8.4.2 观测点	138	10.1 简介	172
8.4.3 控制点	139	10.2 传统电路板测试	172
8.5 易测试性划分	140	10.3 边界扫描体系结构	174
8.6 易测试的电路	145	10.4 测试访问端口	175
8.6.1 C易测试性	146	10.5 寄存器	176
8.6.2 扩充测试	149	10.5.1 边界扫描单元	176
参考文献	150	10.5.2 旁通寄存器	177
习题	151	10.5.3 边界扫描寄存器	177
第9章 路径扫描设计	153	10.5.4 指令寄存器	178
9.1 简介	153	10.5.5 设备识别寄存器	178
9.2 路径扫描设计	153	10.6 TAP控制器	178
9.3 测试码产生	154	10.6.1 控制器状态	178
		10.6.2 指令集	180

10.7 操作模式	181	12.3 缺陷和故障模型	214
10.7.1 正常操作	182	12.3.1 缺陷	214
10.7.2 测试模式操作	182	12.3.2 阵列故障模型	214
10.7.3 测试边界扫描寄存器	184	12.3.3 外围逻辑	217
10.8 边界扫描语言	184	12.4 存储器测试类型	217
10.9 边界扫描设计的代价	184	12.4.1 规格测试	218
10.10 进一步研究	184	12.4.2 特性测试	218
参考文献	185	12.4.3 功能测试	218
习题	185	12.4.4 电流测试	219
第11章 内建自测试	186	12.5 功能测试方案	219
11.1 简介	186	12.5.1 MSCAN	220
11.2 伪随机测试码生成	187	12.5.2 GALPAT 算法	220
11.2.1 线性反馈移位寄存器	187	12.5.3 算法测试序列	220
11.2.2 LFSR结构	188	12.5.4 步进码序列	221
11.2.3 LFSR的数学理论基础	189	12.5.5 棋盘测试	222
11.3 响应压缩	194	12.6 存储器BIST	223
11.3.1 奇偶测试	194	12.7 存储器诊断与维修	225
11.3.2 1-计数	194	参考文献	225
11.3.3 转换计数	195	习题	226
11.3.4 特征分析	196	第13章 FPGA与微处理器的测试	228
11.3.5 空间压缩	199	13.1 简介	228
11.4 抗随机码故障	201	13.2 FPGA	228
11.5 BIST 结构	202	13.2.1 结构	229
11.5.1 BIST结构	202	13.2.2 可编程能力	231
11.5.2 自主测试	202	13.3 FPGA的易测试性	232
11.5.3 循环BIST	203	13.3.1 缺陷与故障	232
11.5.4 BILBO	204	13.3.2 FPGA测试方法	233
11.5.5 随机测试槽	205	13.4 基于RAM的FPGA的测试	233
11.5.6 STUMPS	206	13.4.1 功能测试	233
参考文献	208	13.4.2 I_{DDQ} 测试	235
习题	209	13.4.3 BIST	236
		13.4.4 诊断测试	238
		13.5 微处理器	238
		13.5.1 微处理器模型	239
		13.5.2 微处理器验证	240
		13.6 微处理器的测试	241
		13.6.1 指令集的验证	242
		13.6.2 数据路径的测试	242
		13.7 现代微处理器中的DFT特性	244

第IV部分 特殊结构

第12章 存储器测试	211
12.1 动机	211
12.2 存储器模型	211
12.2.1 功能模型	211
12.2.2 存储器单元	213
12.2.3 RAM组织	213

13.7.1 SUN公司处理器的测试	244
13.7.2 Alpha 21164处理器的测试	245
13.7.3 Intel Pentium Pro的测试	245
13.7.4 AMD K6的测试	247
13.7.5 IBM S/390的测试	247
13.7.6 惠普PA8500的测试	248
参考文献	248
习题	251

第V部分 高级论题

第14章 易测试性综合	253
14.1 简介	253
14.2 易测试性相关内容	253
14.3 综合回顾	254
14.4 高级综合	254
14.4.1 模型编译	255
14.4.2 转化	257
14.4.3 调度	257
14.4.4 分配和绑定	259
14.5 测试综合方法	260
14.5.1 划分	261
14.5.2 可控制性和可观察性	262
14.5.3 反馈回路	263
14.5.4 路径扫描	264

14.5.5 BIST 插入	265
参考文献	268
习题	269
第15章 SOC测试	271
15.1 简介	271
15.2 核的分类	271
15.3 设计与测试流程	272
15.4 内核测试需求	273
15.5 测试体系结构的概念	274
15.5.1 测试数据的源和接受器	275
15.5.2 测试访问机制	275
15.5.3 内核测试包装	276
15.6 测试策略	277
15.6.1 直接访问测试方案	277
15.6.2 应用边界扫描	279
15.6.3 路径扫描的使用	280
15.7 进一步研究	284
15.7.1 虚拟插座接口联盟	284
15.7.2 IEEE P1500 标准	284
参考文献	285
习题	286
附录A 参考书目	287
附录B 缩写词表	292

第I部分 设计与测试

第1章 测试综述

1.1 可靠性与测试

电子系统可靠性已不再局限于那些一旦出现故障将会有灾难性影响的行业，诸如军事、航天及银行业等，其他行业对可靠性和测试技术的需求也与日俱增，比如计算机、通信、消费品及汽车行业等。其原因有以下几点：（1）目前，电子系统已广泛应用于各种工作场所，包括一些苛刻的环境；（2）由于应用范围不断扩大，电子系统的用户不一定都有经验，他们可能会在不经意间有误操作；（3）电子产品尺寸越来越小，同时系统复杂性越来越高，系统速度越来越快，导致了新故障模式的出现。

要获得可靠电子系统的关键条件是确定系统无错的能力[Breuer 1976]。电子系统包含硬件和软件，本书只讨论硬件测试。目前所使用的多数硬件都由数字电路组成，因而本书主要研究数字测试。必须根据说明书对电路进行测试以确保其能够正常的、连续的工作，这种测试主要检测由于生产过程中的缺陷而导致的故障。此外，还可以检测很多由于老化、环境变化、电源脉动等原因造成的故障。测试码生成是一个复杂的问题，通常，用于设计验证的模拟码，由人工或自动测试码生成器（ATPG）产生的码所丰富，以获得更完备的测试集，能够检测电路中的所有故障，而且还可以检验逻辑功能[Abadir 1989]。此外，这些码还可用于使用自动测试设备（ATE）的电路。由于测试过程很复杂，因此使数字电路更容易测试的设计方法已被形式化了。该方法称之为测试设计（DFT）或易测试性设计，这将在后续章节中讨论。其目标是将测试概念嵌入到设计中，使得这些电路更容易控制和观察。

可靠性的另一方面是系统根据要求可靠运行的能力。它要求系统是容错的。容错是个极其丰富的领域，它包含了数字测试，但本书中不会涉及很多。要进一步探讨该话题的话，建议查阅诸如[Siewiorick 1982]和[Johnson 1989]这样的文献。

为了更好地理解验证和测试的作用和范畴，以及这两个过程之间的关系，本章将首先介绍数字设计过程，然后给出本书所讨论内容的纲要。

1.2 设计过程

随着晶体管体积的不断缩小，设备密度和设计复杂性稳定地增长。目前晶体管的密度已达到上亿个，要管理这样的复杂系统，自然应该逐级设计。除了复杂性之外，数字系统的生命周期有时会比设计周期短。为了保持其在电子领域内的竞争力，供应商需要提高设计者的效率，并减少进入市场的时间。

设计过程也应该随着设计者可利用的技术和CAD工具的发展而不断变化。设计过程可以分为三个主要阶段：系统设计、逻辑设计和物理设计。随着自动化程度的提高，实现三个阶段中的任何一个所需要的时间正逐步减少，如图1-1所示。每个设计阶段都包括设计和验证，然后转入下一个阶段。通常用模拟方法进行验证，但最近，形式化验证正受到越来越多的重视。

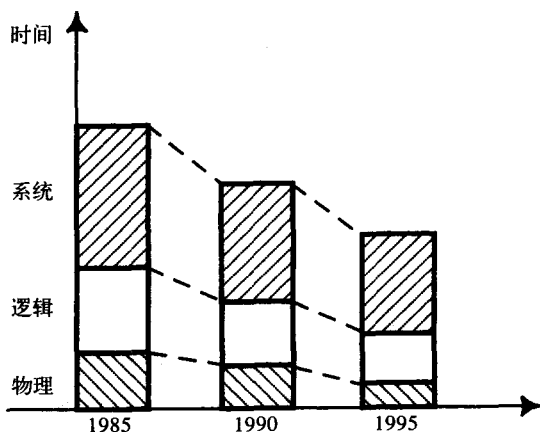


图1-1 各阶段的设计时间

亚微型技术的发展使得集成电路（IC）设计者能够设计更复杂的系统，甚至可以在一个芯片上实现整个系统。这种新的片上系统（SOC）范例已经改变了设计和测试方法。要提高设计生产能力，缩短上市时间，在SOC设计中，重用以前设计的模块正变得越来越普及。这被称之为基于内核的设计。这种可重用方法并不局限于内部设计，而是可以扩展到他人设计的模块，这样的模块被称做嵌入内核。

图1-2显示了一个SOC实例。它包含不同的内核（矩形块）和用户定义逻辑（UDL）。内核可以是处理器、DSP或RAM等。UDL组件将扩展系统中不同的内核组合在一起。内核可以是预先设计好的，并可以在不同级中加以描述，比如从说明书到硬件描述语言再到版图。由于设计SOC很复杂，正如图1-1所表现的那样，因此，系统设计阶段所花时间要比其他设计阶段长。本书将在第15章讨论SOC的设计和测试。在本章后面部分，讨论如何根据说明书来完成内核或UDL组件的设计。

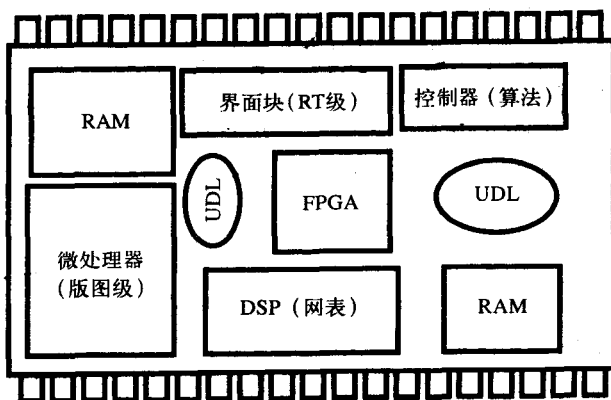


图1-2 片上系统

在过去，硬件设计主要是在纸上完成的，现在是用硬件描述语言（HDL）来描述。使用HDL进行设计有几个优点，其中最重要的优点是可进行复杂性管理并缩短了设计周期。目前，设计可以从行为级直接进入寄存器传送级（RTL），再到如图1-3所示的逻辑级，最后进行布局设计。这些自动转换被称做设计综合。综合和模拟工具都使用了组件库。例如，考虑对一个设计进行逻辑描述，假设该设计包含基本逻辑门，如NAND、NOR等，这些组件构成了逻辑库。每个逻辑门都有一个依赖于过程的版图视图，各个门的版图构成了物理库。设计人员有权使用经销商提供的标准单元库，而且他们可以对库进行扩充，甚至于开发自己的库。这种从门级到物理级的转换被称做技术映射。

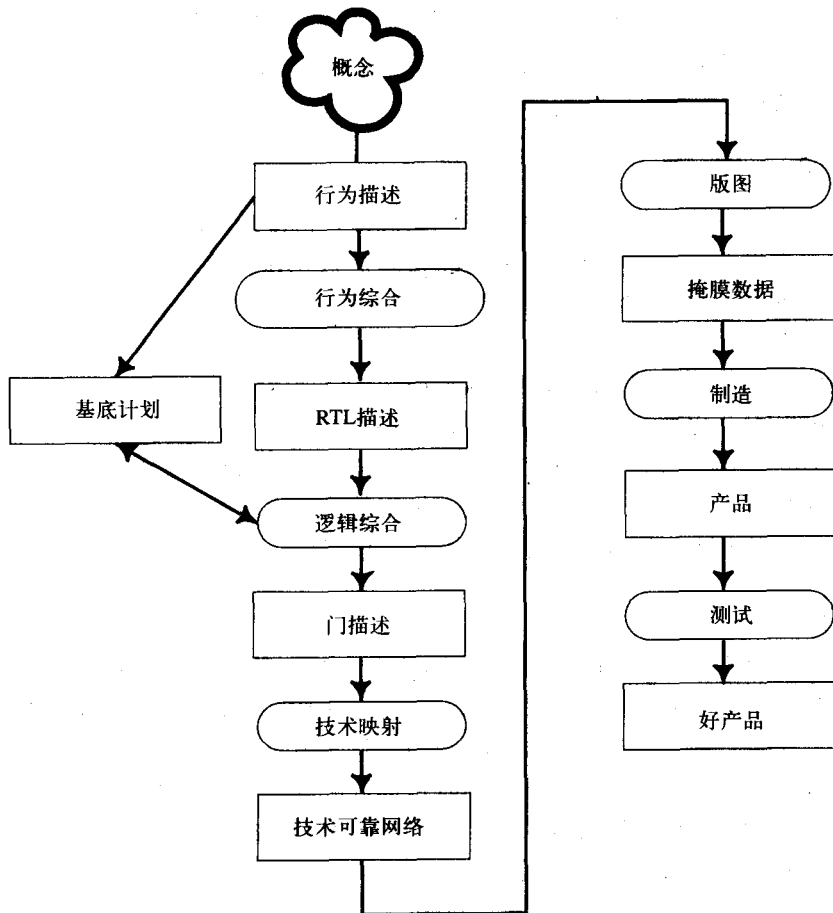


图1-3 设计周期

从一级步进到另一级时，第一级都要重复使用模拟技术，一直到设计满意为止。一旦版图完成，就可以提取诸如负载电阻和电容等参数了。利用这些信息，可以验证电路的时序。目前，深亚微技术中的参数提取正变得越来越重要。

版图掩模工艺可以用来制作晶片压模，完成制作后，晶片被测试并标记有缺陷的压模，该过程被称作晶片分类。好的压模被打包，然后利用ATE设备进行测试。由于设计和测试之间的关系密切，因此本书用了两章的篇幅来介绍这一主题及设计表示。

1.3 验证

在电路实现之前进行的测试称做设计验证。没有哪个工程师能把不经过设计验证的设计拿去给铸造厂。问题不在于是否要验证,而在于如何验证好,使得产品设备能够完全达到设计说明书的要求。在设计阶段,电路可有几种描述,如图1-3所示。图上四种描述——行为级、RTL级、门级和版图级是从不同的角度对同一电路的描述。从一个设计阶段到另一个设计阶段过渡时,可能会产生一些故障。这些故障可能是由于CAD工具或人为误操作所导致的。在每一阶段,验证设计就是为了证明它是来自前一阶段的同一设计,而且符合设计说明书。目前,对于硬件验证来说,模拟器是最流行的工具。虽然到目前为止,在形式验证上已进行了广泛的研究,但几乎没有实际的工具能证明基于形式验证的设计的正确性。穷举模拟就等价于形式验证,然而,对目前的复杂电路来说,这样的模拟并不可行。用于验证设计的两类模拟为:功能模拟和时序模拟。

1.3.1 功能模拟

在功能模拟中,设计人员可以验证设计的正确性,但没必要以实际运行速度来进行,就是说,不必包含功能单元时延(或者至多是常量时延),功能单元时延通常称做零(或单元)时延。必须注意的是(1)检查是否每一块都执行了指定的功能;(2)在确定设计方案之前尝试一下其他的方法。

模拟器将电路描述转换成相应的内部数据结构,以便于解释。例如,一个电路图被映射为一个网表(netlist),它是所有门及其链接的数据结构列表。第3章将介绍各种电路表示及其对CAD工具性能的影响。设计人员准备了输入集、验证码,并将它们应用于电路,然后通过输出来检验其正确性。之后,如果设计人员对输出结果满意,则进行转换。使用同一组测试向量对实际设备进行测试,以确保所设计的电路实现正确,这是一种很好的实践方式。

1.3.2 时间模拟

模拟结果是将设计原型投入批量生产的坚实基础。实际的原型验证又提供了更多的保证,因为它可以使依赖于工艺的参数具体化。幸运的是,目前的CAD工具包括了参数提取,允许在实际加工过程中进行模拟。时间验证可以与功能验证协同完成或者独立进行,还可为相关各门(或在HDL模型情况下的功能单元)的时延进行赋值。通常,赋予的时延很少。时延是设计时所使用的库的一部分,还可以修改。这类模拟中不包含网络时延,因而,时间验证也是不完备的。只有当设计实施时,才能赋予各门及其连接实际的时延。此外,负载可以评估并赋予输出端。

1.4 测试

一旦在硅片上实现了设计,就可以应用适当的激励并检查输出来验证其正确性。然而,测试并不是对设计的验证,而是对制造正确性的验证。有两类主要的测试:参数测试和功能测试。参数测试主要考虑电路参数,比如电流和电压的测试。除非保证有极小的电压和电流,否则就没有必要进一步测试了,而功能测试就是本书所要讨论的内容。

正如前面所提到的,测试的目的就是为了证明制造的IC无错。首先,需要定义故障。早期的数字电路测试技术主要关心功能验证。这里要谈到R.Eldred在1959年8月[Eldred

1959]ACM会议论文上所提出的考虑电路结构的测试方法。在该论文的开头写道：“为了保证测试程序成功运行，以确保计算系统没有故障组件，常规的测试条件应该从组件级本身来设计，而不是从程序级上设计”。

为了更好地管理测试，使电路中的故障表现为连接电路各组件节点上的逻辑值或电量值是非常重要的，即通过一个逻辑值来表示故障模式。这就等于在逻辑级上通过模型来表示物理故障。如果这种映射是一对一的，则会有大量模型表示。作为模型，不一定要精确地表示故障，但在检测故障时却很有用。例如，最常见的故障模型是线上的单固定型故障SSA(single stuck-at line)，它并不能精确地表示所有实际的物理故障。尽管它现在很流行，但对目前的电路和技术来说，固定型故障已不再满足需要了。随着MOS技术的出现，显然在这一技术中需要额外的故障模型来表示更多的综合故障模式[El-Ziq 1981]。故障模型是第2章所讨论的内容。

这里已经表明了使用测试码模式证明产品无错的数字测试可以在实际IC上完成。这种测试观点意味着测试生成可以在门级设计中完成，而且过去就是这样做的。目前，设计周期可以和测试周期并行进行，如图1-4所示。因此，每个设计验证步骤都能和测试周期中的一个部分相对应。例如，在逻辑级，测试生成所提供的测试码可用故障模拟器来验证。这个趋势也可扩展到比较抽象的更高级设计，易测试性也可在同样的级别上进行。在RTL级和行为级上尝试测试模式生成已有了相关报道。然而，在这些抽象的设计级别上，还有其他的评估电路易测试性的方法，而不用实际开发模式，一般称之为易测试性度量，这在本节的后面进行了描述，并将在第8章进行详细介绍。

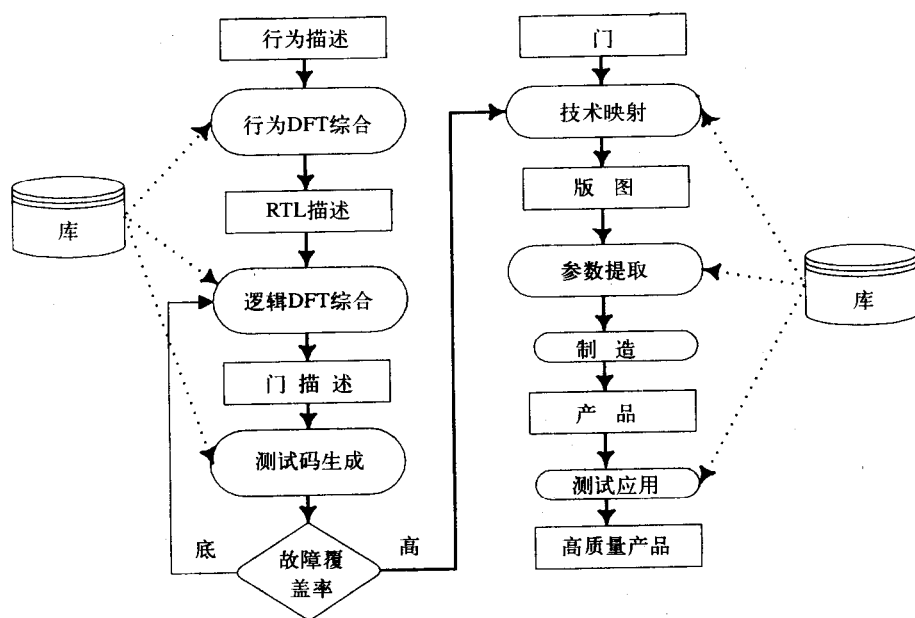


图1-4 DFT周期

和设计密切相关的另外一个基本的测试部分是易测试性设计实践。该方法提倡将嵌入式测试纳入电路中，从而使得测试更方便、更有效。这里的方便、有效是指测试码生成和测试应用处理。测试正成为最优化设计的一个因素。设计人员通常努力想达到一种最佳的设计：高速、低功率设计，并占用尽可能小的区域。然而，根据产品的使用来看，设计人员通常使这三个属