

21世纪高等院校电子信息类规划教材  
21st Century University Planned Textbooks of Electronic Information

# 超大规模 集成电路设计

曲英杰 方卓红 编著

Design of Very Large  
Scale Integrated Circuit

 人民邮电出版社  
POSTS & TELECOM PRESS

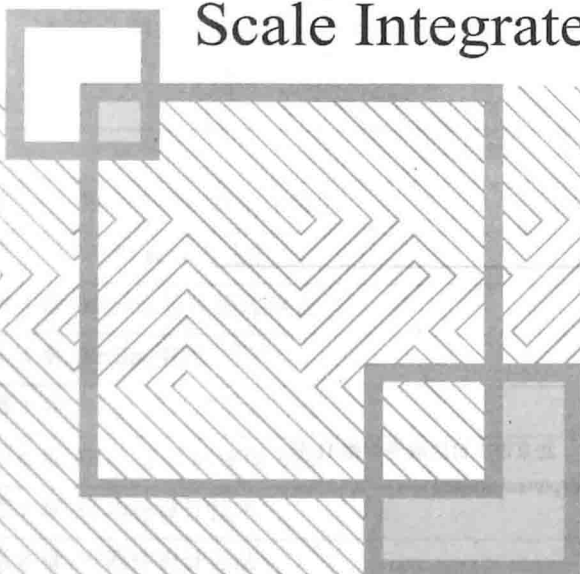
  
高校系列

纪高等院校电子信息类规划教材  
Jury University Planned Textbooks of Electronic Information

# 超大规模 集成电路设计

曲英杰 方卓红 编著

Design of Very Large  
Scale Integrated Circuit



人民邮电出版社  
北京



高校系列

## 图书在版编目 (CIP) 数据

超大规模集成电路设计 / 曲英杰, 方卓红编著. --

北京: 人民邮电出版社, 2015. 2

21世纪高等院校电子信息类规划教材

ISBN 978-7-115-38382-2

I. ①超… II. ①曲… ②方… III. ①超大规模集成电路—电路设计—高等学校—教材 IV. ①TN470.2

中国版本图书馆CIP数据核字(2015)第017503号

## 内 容 提 要

本书根据超大规模集成电路设计的工程需要来确定知识结构, 内容涵盖了超大规模集成电路设计流程中的各个知识点, 系统地介绍了超大规模集成电路的设计思想、原理、方法和技术。主要内容包括数字集成电路设计概述、VLSI 设计方法学、Verilog 硬件描述语言、Verilog HDL 逻辑设计方法、VLSI 设计的验证方法、EDA 工具的使用方法、低功耗设计技术、可测性设计方法, 以及 3 个设计实例。全书设计实例丰富且介绍详尽, 通过学习, 学生能够深入了解各个设计环节, 加深对设计方法的理解, 提高其工程实践能力。

本书可作为高等学校电子信息类专业本科生和研究生的教材, 也可供相关的工程技术人员参考。

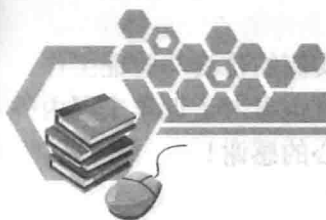


- 
- ◆ 编 著 曲英杰 方卓红  
责任编辑 张孟玮  
执行编辑 税梦玲  
责任印制 沈 蓉 彭志环
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京铭成印刷有限公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 24  
字数: 620 千字
- 

定价: 59.00 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315



超大规模集成电路设计技术是信息产业的核心技术，关系到国家的安全和经济的发展，体现了一个国家的核心竞争力和创新能力。目前，我国集成电路设计产业发展迅速，人才缺口较大，很多高校的电子类专业陆续开设了“超大规模集成电路设计方法”这一课程。为了提高我国超大规模集成电路设计人才的培养质量，为我国集成电路设计产业的发展提供人才支持，编者结合自身教学经验，编写了此书。

本书的指导思想是：面向学科前沿，面向产业需求，面向工程实践，在内容的选择和组织上力求体现出系统性、完备性、先进性和实用性。基于上述指导思想，本书首先介绍了数字集成电路设计的历史、现状和发展趋势，以及相关的基本概念、思想和方法，然后介绍了目前产业界流行的超大规模集成电路设计流程，之后以该设计流程为主线，详细介绍流程中各个环节所用到的设计思想、原理、方法和技术。

本书内容新颖，紧跟学科前沿，将新的理论研究成果和先进的实用技术纳入其中；知识结构系统完备，内容涵盖了超大规模集成电路设计流程中的各个知识点；设计实例丰富且介绍详尽，每个实例都来自作者的科研项目；注重总结和提炼具有普遍指导意义的设计思想和方法，引导学生重视对一般性的设计思想、设计方法的理解和掌握，而非仅仅局限于某些具体问题的解决，从而提高学生分析问题和解决问题的能力；在内容的组织编排方面突出逻辑性和条理性，遵循循序渐进的原则，符合读者的学习习惯。

建议采用本书的授课学时为 80 课时，其中理论课时为 48 课时，实验课时为 32 课时。具体课时分配见下表。

授课内容	课时分配
第 1 章 数字集成电路设计概述	4 课时
第 2 章 VLSI 设计方法学	8 课时
第 3 章 Verilog 硬件描述语言	8 课时
第 4 章 Verilog HDL 逻辑设计方法	6 课时
第 5 章 VLSI 设计的验证方法	4 课时
第 6 章 EDA 工具的使用方法	8 课时
第 7 章 低功耗设计技术	2 课时
第 8 章 可测性设计方法	2 课时
VLSI 设计实例（可从第 9~11 章 3 个设计实例中选择 1 个讲授）	6 课时
实验（可从第 9~11 章 3 个设计实例中选择 1 个作为实验内容，将设计实例划分为体系结构设计、各个子模块的 RTL 代码设计与功能仿真、系统集成与功能仿真、基于 FPGA 进行综合与布局布线、时序仿真、FPGA 实现与测试等多个实验）	32 学时



本书由曲英杰教授编著，方卓红老师参与完成了书中全部设计实例的仿真与验证工作。在本书的编写过程中得到了青岛科技大学教材建设项目的支持，同时也参考了近年来国内外出版的多本同类教材和相关网络资料，在此一并向相关人员表示衷心的感谢！

限于作者水平，书中难免有不当之处，敬请广大读者批评指正！

编者

E-mail: quyj\_qd@163.com

2014年11月

本书共分11章，第1章介绍超大规模集成电路的发展概况、分类及应用；第2章介绍超大规模集成电路的制造工艺；第3章介绍超大规模集成电路的封装技术；第4章介绍超大规模集成电路的测试技术；第5章介绍超大规模集成电路的可靠性设计；第6章介绍超大规模集成电路的功耗管理；第7章介绍超大规模集成电路的电磁兼容设计；第8章介绍超大规模集成电路的散热设计；第9章介绍超大规模集成电路的电源设计；第10章介绍超大规模集成电路的信号完整性设计；第11章介绍超大规模集成电路的时钟设计。

章节标题	主要内容
第1章	超大规模集成电路的发展概况、分类及应用
第2章	超大规模集成电路的制造工艺
第3章	超大规模集成电路的封装技术
第4章	超大规模集成电路的测试技术
第5章	超大规模集成电路的可靠性设计
第6章	超大规模集成电路的功耗管理
第7章	超大规模集成电路的电磁兼容设计
第8章	超大规模集成电路的散热设计
第9章	超大规模集成电路的电源设计
第10章	超大规模集成电路的信号完整性设计
第11章	超大规模集成电路的时钟设计

青岛科技大学 (101) 11555115





<b>第 1 章 数字集成电路设计概述</b> .....	1
1.1 数字集成电路的发展历史与现状 .....	1
1.1.1 机械式计算机的启蒙时代 .....	1
1.1.2 电子技术和半导体技术的诞生和发展 .....	1
1.2 现代数字 IC 设计方法的发展 .....	5
1.2.1 自底向上的设计方法 .....	5
1.2.2 自顶向下的设计方法 .....	5
1.2.3 自顶向下与自底向上相结合的设计方法 .....	6
1.3 数字 IC 前端设计语言及后端设计软件 .....	6
1.3.1 Verilog 硬件描述语言 .....	7
1.3.2 VHDL 设计语言 .....	7
1.3.3 验证和验证语言 .....	7
1.3.4 数字 IC 设计后端 EDA 工具 .....	8
1.4 数字 IC 的设计模式 .....	9
1.4.1 全定制设计模式 .....	9
1.4.2 标准单元设计模式 .....	9
1.4.3 门阵列设计模式 .....	9
1.4.4 宏模块设计模式 .....	9
1.4.5 FPGA 设计模式 .....	9
1.4.6 不同设计模式的比较 .....	10
1.5 数字 IC 设计面临的挑战 .....	10
1.5.1 工艺极限的挑战 .....	10
1.5.2 投资风险的挑战 .....	11
1.5.3 IC 工程师面临的挑战 .....	11
1.5.4 项目管理上的挑战 .....	11
1.6 集成电路的分类 .....	11
1.6.1 按用途分类 .....	11
1.6.2 按集成度分类 .....	12
1.6.3 按设计与制造过程分类 .....	12
1.7 集成电路设计与制造相关的常用术语和基本概念 .....	13
1.8 集成电路设计质量评价 .....	14
<b>第 2 章 VLSI 设计方法学</b> .....	16
2.1 VLSI 设计流程简介 .....	16
2.2 系统体系结构设计 .....	17
2.2.1 系统体系结构设计的内容及方法 .....	17
2.2.2 系统体系结构设计实例 .....	18
2.3 RTL 代码编写 .....	51
2.4 RTL 代码功能仿真 .....	52
2.5 综合优化 .....	54
2.6 可测性设计 .....	57
2.7 后端布局布线 .....	58
2.8 时序仿真 .....	59
2.9 静态时序分析与时序收敛 .....	59
2.9.1 静态时序分析 .....	59
2.9.2 时序收敛 .....	59
2.10 CMOS 工艺选择 .....	60
2.11 IC 产业的变革及对设计方法的影响 .....	61

<b>第 3 章 Verilog 硬件描述语言</b> .....	62	4.6.1 多周期处理机 Verilog RTL 代码 .....	131
3.1 引言 .....	62	4.6.2 多周期处理机测试 代码 .....	135
3.2 Verilog HDL 基本结构 .....	63	4.6.3 多周期处理机功能 仿真 .....	136
3.2.1 简单的 Verilog HDL 例子 .....	63	4.7 可综合的 Verilog RTL 设计 .....	137
3.2.2 Verilog HDL 的基本 结构 .....	64	4.7.1 可综合的组合电路 设计 .....	137
3.2.3 逻辑功能定义 .....	64	4.7.2 可综合的时序电路 设计 .....	138
3.2.4 关键字 .....	66	4.8 代码书写风格 .....	139
3.2.5 标识符 .....	66	<b>第 5 章 VLSI 设计的验证方法</b> .....	140
3.2.6 编写 Verilog HDL 源代码的标准 .....	66	5.1 VLSI 设计验证的原理与 方法 .....	140
3.3 数据类型及常量、变量 .....	68	5.2 软件仿真举例 1: RSA 加密 处理器仿真 .....	141
3.4 运算符及表达式 .....	72	5.3 软件仿真举例 2: 基于 USB 的 RSA 加密处理器的功能 仿真 .....	146
3.5 语句 .....	75	5.4 软件仿真举例 3: AES 加密 处理器的时序仿真 .....	150
3.6 赋值语句和块语句 .....	76	5.5 硬件仿真举例: 基于 USB 的 AES/RSA 加密处理器的硬件 仿真 .....	153
3.7 条件语句 .....	79	5.6 验证平台编码风格 .....	158
3.8 循环语句 .....	81	5.7 验证平台模块设计 .....	158
3.9 结构说明语句 .....	84	5.8 验证平台结构设计 .....	162
3.10 编译预处理语句 .....	89	5.9 断言 .....	163
3.11 语句的顺序执行与并行 执行 .....	90	5.10 验证质量评估 .....	164
3.12 不同抽象级别的 Verilog HDL 模型 .....	92	<b>第 6 章 EDA 工具的使用方法</b> .....	165
3.13 设计技巧 .....	94	6.1 ModelSim 的使用方法 .....	165
<b>第 4 章 Verilog HDL 逻辑设计 方法</b> .....	96	6.1.1 ModelSim 概览 .....	165
4.1 基本组合电路的设计方法 .....	96	6.1.2 基本仿真流程 .....	167
4.2 基本时序电路设计 .....	101	6.1.3 ModelSim 工程 .....	173
4.3 同步状态机的设计方法 .....	104	6.1.4 基于多库的仿真 .....	179
4.4 存储模块设计 .....	123		
4.5 复杂数字系统的逻辑设计 .....	128		
4.5.1 算法状态机图 .....	128		
4.5.2 数据通道/控制器 划分 .....	129		
4.5.3 复杂数字系统的设计 方法 .....	130		
4.6 复杂数字系统设计举例: 多周期处理机设计 .....	131		



6.1.5	在波形窗口中查看仿真结果 .....	183	8.3.3	扫描设计的注意事项 .....	270
6.1.6	利用 ModelSim 进行时序仿真 .....	188	8.4	边界扫描电路设计 .....	271
6.2	Quartus II 的使用方法 .....	190	8.4.1	边界扫描简介 .....	271
6.2.1	可编程逻辑设计流程 .....	190	8.4.2	边界扫描电路结构 .....	272
6.2.2	设计输入 .....	201	8.4.3	TAP 控制器及指令集 .....	272
6.2.3	综合 .....	209	8.4.4	基于 BSD Compiler 的边界扫描电路设计方法 .....	275
6.2.4	仿真 .....	212			
6.2.5	布局布线 .....	217			
6.2.6	基于块的设计 .....	223	<b>第 9 章 VLSI 设计实例 1: AES 密码处理器设计</b> .....	<b>280</b>	
6.2.7	时序分析 .....	227	9.1	AES 算法描述 .....	280
6.2.8	时序逼近 .....	232	9.1.1	数学预备知识 .....	280
6.2.9	编程与配置 .....	236	9.1.2	符号和习惯用语 .....	281
6.2.10	调试 .....	240	9.1.3	AES 加密算法 .....	283
6.2.11	工程更改管理 .....	244	9.1.4	AES 解密算法 .....	286
6.2.12	系统级设计 .....	248	9.1.5	密钥扩展 .....	289
6.2.13	软件开发 .....	251	9.2	AES 密码处理器的体系结构设计 .....	290
<b>第 7 章 低功耗设计技术</b> .....	<b>254</b>		9.2.1	AES 密码处理器框图及外部信号说明 .....	290
7.1	低功耗设计的背景和意义 .....	254	9.2.2	AES 密码处理器模块结构图 .....	291
7.2	低功耗设计技术的发展趋势 .....	255	9.2.3	AES 密码处理器各子模块设计方案 .....	291
7.3	常用的低功耗设计技术介绍 .....	257	9.3	AES 密码处理器的 Verilog 模型设计 .....	301
<b>第 8 章 可测性设计方法</b> .....	<b>261</b>		9.4	AES 密码处理器的功能仿真 .....	316
8.1	可测性设计的背景及意义 .....	261	9.4.1	密钥扩展仿真结果 .....	316
8.2	可测性设计的基本概念与方法 .....	261	9.4.2	加密仿真结果 .....	317
8.2.1	常用缩略语解释 .....	261	9.4.3	解密仿真结果 .....	318
8.2.2	DFT 的常用方法 .....	262	9.5	基于 FPGA 的 AES 密码处理器的实现与测试 .....	319
8.3	基于 ATPG 的扫描测试 .....	266	9.5.1	基于 FPGA 的 AES 密码处理器的综合与时序仿真 .....	319
8.3.1	扫描测试的基本原理 .....	266			
8.3.2	扫描测试的主要阶段 .....	268			
8.3.3	扫描测试的基本时序 .....	269			



9.5.2 基于 FPGA 的 AES 密码处理器的实现 与测试 .....	321	第 11 章 VLSI 设计实例 3—— RSA 密码处理器设计 .....	359
<b>第 10 章 VLSI 设计实例 2: SMS4 密码处理器设计 .....</b>	<b>324</b>	11.1 RSA 算法描述 .....	359
10.1 SMS4 密码算法简介 .....	324	11.2 RSA 算法硬件实现原理 .....	360
10.1.1 说明与定义 .....	324	11.3 RSA 密码处理器体系结构 设计 .....	361
10.1.2 SMS4 加解密过程 .....	325	11.3.1 RSA 模块框图及外 部信号说明 .....	361
10.1.3 SMS4 密钥的扩展 过程 .....	325	11.3.2 RSA 密码处理器总 体结构设计方案 .....	362
10.2 体系结构设计 .....	326	11.3.3 Montgomery 模乘运算 子模块设计方案 .....	363
10.2.1 基于单轮循环迭代的 设计 .....	326	11.3.4 模数/明密文/参数 AC 寄存器堆设计 方案 .....	364
10.2.2 基于全流水线的 设计 .....	328	11.3.5 密钥寄存器堆设计 方案 .....	364
10.2.3 基于流水线与循环相 结合的设计 .....	331	11.3.6 MM 输入寄存器设 计方案 .....	365
10.3 RTL 级模型建立与功能 仿真 .....	340	11.3.7 RSA 控制模块设 计方案 .....	365
10.3.1 基于单轮循环迭代的 RTL 级模型建立与 仿真 .....	340	11.4 RSA 密码处理器 RTL 级设 计及仿真 .....	368
10.3.2 基于全流水线的 RTL 级模型建立与 仿真 .....	343	11.4.1 系统控制模块 .....	369
10.3.3 基于流水线与循环相 结合的 RTL 级模型 建立与仿真 .....	345	11.4.2 Montgomery 模乘 模块 .....	370
10.4 FPGA 实现与测试 .....	348	11.4.3 RSA 系统总模块 .....	371
10.4.1 Cyclone 器件 简介 .....	348	11.5 RSA 密码处理器的综合 .....	372
10.4.2 基于 Quartus II 的 综合 .....	349	11.5.1 RTL 级逻辑图 .....	372
10.4.3 结合 ModelSim 的时 序仿真 .....	353	11.5.2 综合结果 .....	372
10.4.4 基于 Altera-cyclone 芯片的实现 .....	357	11.6 RSA 密码处理器的 FPGA 实现与测试 .....	373
		11.6.1 RSA 密码处理器的 FPGA 实现 .....	373
		11.6.2 RSA 密码处理器的 测试 .....	373
		<b>参考文献 .....</b>	<b>376</b>



# 第 1 章

## 数字集成电路设计概述

本章主要介绍集成电路设计相关的基本概念和知识,内容包括数字集成电路的发展历史与现状、现代数字 IC 设计方法的发展、数字 IC 前端设计语言及后端设计软件(EDA)、数字 IC 的设计模式、数字 IC 设计面临的挑战、集成电路的分类、集成电路相关的基本概念、集成电路设计质量评价等。

### 1.1 数字集成电路的发展历史与现状

半个多世纪以来,数字集成电路技术的发展成为科学与技术各个方面进步的巨大动力,并且影响到了人类活动的各个方面。电子计算机的发展和广泛应用就是数字集成电路技术应用的一个非常成功和典型的例子,每一次计算机性能的提高都离不开数字集成电路技术进步的推动,而计算机性能的提高和应用的普及反过来又促进了数字集成电路技术的快速发展,因此,数字集成电路的发展历史与计算机的发展历史密切相关。

#### 1.1.1 机械式计算机的启蒙时代

在电子元器件发明之前,人们就开始探索用机器代替人进行计算的可能性。最初出现的计算器就是中国古人发明的算筹和算盘,此后欧洲陆续发明了计算尺和手摇式计算器,实现了超越函数的计算和一些开方运算。法国数学家帕斯卡(Pascal)发明的钟表式齿轮计算机,是机械式计算机的初级阶段,此后莱布尼茨乘法器、巴贝奇微分器也相继问世。图 1.1 为巴贝奇微分器,它是一个机械计算装置,能执行加、减、乘、除基本运算(十进制),分“存放”和“执行”两个周期序列,共有 25 000 个部件,成本为 17 470 英镑。

1703 年,德国数学家莱布尼茨(Leibniz)的论文《谈二进制算术》发表在《皇家科学院论文集》上。19 世纪,英国数学家布尔(Boole)运用代数方法研究逻辑学,1844 年发表了著名论文《关于分析中的一个普遍方法》。这些数学理论为日后的数字系统设计和电子计算机的发明奠定了坚实的科学基础。

#### 1.1.2 电子技术和半导体技术的诞生和发展

20 世纪 40~70 年代,电子技术和半导体工艺技术的突飞猛进为数字设计的发展提供了新的舞台,数字设计技术得到了快速

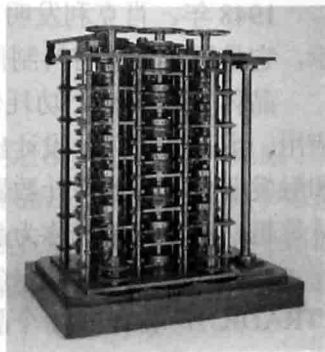


图 1.1 巴贝奇微分器

发展和广泛应用。

### 1. 电子管时代

美国发明家佛斯特 (Lee de Forest) 于 1906 年发明了电子管, 随即用它来放大无线电信号和声音信号。1909 年, 美国贝尔 (Bell) 购买了他的专利, 经过改进, 用于长距离电线电信号的放大。之后以电子管为核心器件陆续发明了许多产品: 电子管收音机、电子管录放机、电子管电视机、唱片机、无线电发报机、电子计算机等。

电子管是封装在玻璃外壳内的一种电真空器件, 如图 1.2 所示。用电子管可以设计出实现反相功能的反相器线路, 在此基础上, 再实现出计算机使用的全部组合逻辑线路, 诸如加法器、译码器等线路, 和触发器、寄存器、计数器等各种时序逻辑线路。用电子管线路实现的属于电子管计算机。

世界上第一台电子数字计算机 ENIAC (Electronic Numerical Integrator And Calculator), 1946 年由美国宾夕法尼亚大学研制, 字长 12 位, 运算速度 5000 次/秒, 使用 18800 个电子管、1500 个继电器, 功耗 150kW, 占地 170m<sup>2</sup>, 重达 30 吨, 造价 100 万美元, 如图 1.3 所示。



图 1.2 电子管

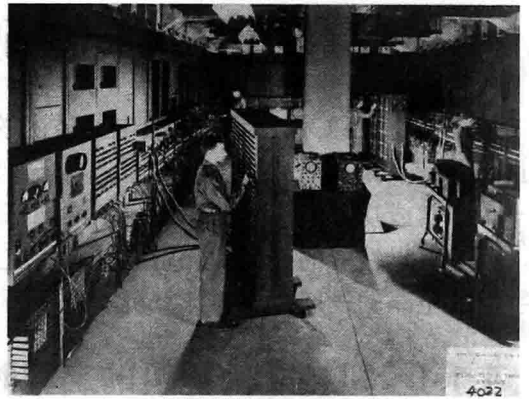


图 1.3 世界上第一台电子数字计算机 ENIAC

### 2. 晶体管时代

电子管体积大、功耗高、价格贵、易破碎, 为了改进这些缺陷, 人们开始寻找电子管的替代元件。

1947 年, 贝尔实验室肖克利领导的半导体研究小组发明了点接触型晶体管, 如图 1.4 所示。

1948 年, 肖克利发明了面接触型晶体管, 这就是今天仍在使用的晶体管结构, 如图 1.5 所示, 它是用半导体材料制作的、封装在一个金属壳内的带有 3 个管脚的小器件。

晶体管体积小、功耗低、性能更加稳定, 逐步取代了电子管的地位, 从而得到了广泛的应用。用晶体管可以设计出实现反相功能的反相器线路, 在此基础上, 可实现组合逻辑线路, 和触发器、寄存器、计数器等时序逻辑线路, 因此利用晶体管作为基本元件可以设计实现电子计算机, 这种计算机称为晶体管计算机。

1954 年, 美国贝尔实验室研制出第一台使用晶体管线路的计算机, 取名“崔迪克” (TRADIC), 装有 800 个晶体管。1958 年, 美国 IBM 公司制成第一台全部使用晶体管的计算机 RCA501 型。1964 年, 中国制成了第一台全晶体管电子计算机 441-B 型。

晶体管计算机性能和可靠性提高, 体积和价格下降, 外设和软件也越来越多, 并且高级



程序设计语言应运而生，从而使计算机工业迅速发展。

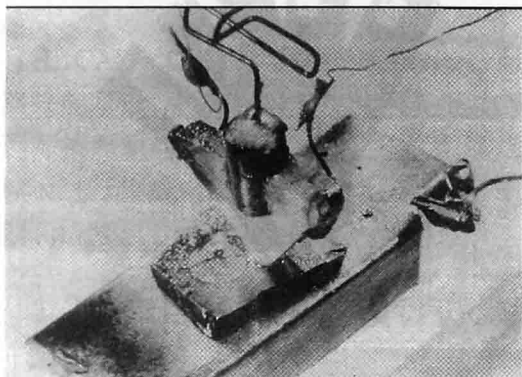


图 1.4 点接触型晶体管

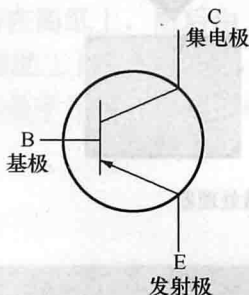
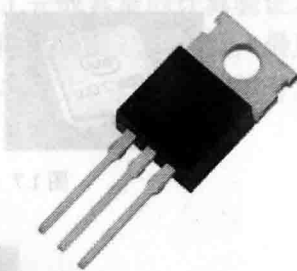


图 1.5 面接触型晶体管结构和实物



尽管用晶体管替代电子管使计算机面貌焕然一新，但随着对计算机性能的追求，新计算机包含的晶体管数量已从一万个左右骤增到数十万个，人们需要将这些晶体管、电阻等元件都焊接到一块电路板上，再由一块块电路板通过导线连接到一台计算机上。其复杂工艺严重影响了制造计算机的生产效率，使计算机可靠性很低。

### 3. 集成电路时代

电子管的很多缺点延续到分立晶体管身上，复杂的连线导致电路系统设计复杂，而且体积、功耗比较大，电子产品的电路成本和使用成本都比较高。克服晶体管的这些缺陷成为半导体工艺工程师后来追逐的主要目标，从而导致了集成电路的发明。

集成电路 (integrated circuit, IC) 是指以半导体晶体材料为基片，经加工制造，将元件和互连线集成在基片内部、表面或基片之上，再用一个管壳将其封装起来，形成具有某种电子功能的微型化电路。如图 1.6、图 1.7、图 1.8 所示。有时也把集成电路称为芯片。

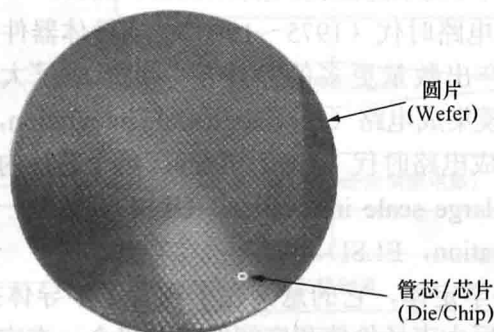


图 1.6 晶圆和芯片

1958 年，美国德州仪器 (Texas Instruments) 的工程师基尔比制造出第一块 IC，集成了 1 个晶体管、1 个电容、1 个电阻，如图 1.9 所示。1959 年，仙童 (Fairchild) 半导体科学家诺伊斯发明了可制造性更强的 IC 设计技术，从而开启了集成电路快速发展的时代，集成电路的集成度和性能按照摩尔定律迅速提高，即单片集成电路上可容纳的晶体管数目，大约每隔 18 个月便会增加一倍，性能将提升一倍。

按照集成度划分，集成电路的发展经历了中小规模集成电路时代、大规模和超大规模集成电路时代、甚大规模和极大规模集成电路时代等。

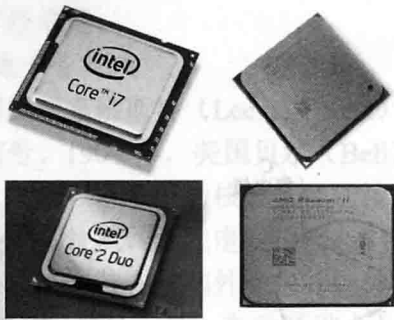


图 1.7 微处理器

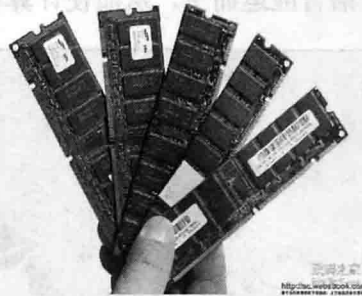


图 1.8 存储器芯片

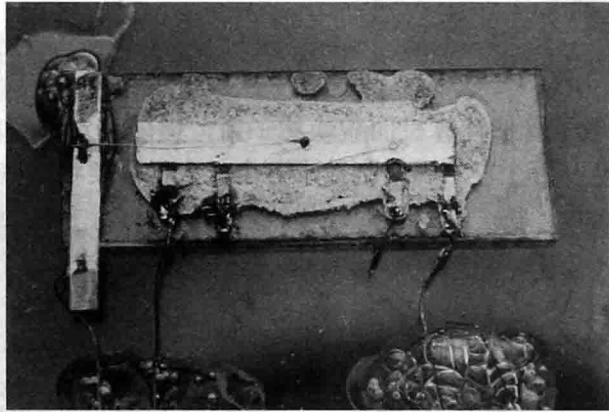


图 1.9 基尔比制造的第一块 IC

中小规模集成电路时代（1964~1975）：此时集成到一个芯片内的晶体管数量还相当有限，实现的还只限于简单的、完成基本处理功能的组合逻辑门一级的电路和简单的触发器、寄存器之类的电路，故被称为中、小规模集成电路（MSI、SSI）。

大规模和超大规模集成电路时代（1975~1990）：半导体器件生产工艺的改进，使得在一片半导体基片上，可以生产出数量更多的晶体管，就形成了大规模集成电路（large scale integration, LSI）和超大规模集成电路（very large scale integration, VLSI）。

甚大规模和极大规模集成电路时代（1990 至今）：单个芯片内的晶体管数量达到百万个时被叫做甚大规模电路（ultra large scale integration, ULSI），达到一亿个时被叫做极大规模电路（extremely large scale integration, ELSI）。

摩尔定律并不是什么科学定律，它的意义在于预言了半导体技术的发展前景，最重要的是它还指出了半导体技术的巨大市场价值和广阔的商业机会。在它的影响下，全世界为半导体技术投入了巨大的资金、人力、物力，半导体技术吸引了最杰出的人才，成为人类史上最辉煌的、发展最快的科学技术领域。一直到今天，摩尔定律依然以“更快、更小、更便宜”的核心思想指导着芯片设计行业的发展。

今天，在单片集成电路上可以集成多达十亿只晶体管，我们拥有非常丰富的设计资源，可以在这样的平台上进行自由的、复杂的、高速的数字系统设计。可以毫不夸张地说，只要能用数学算法和软件算法描述清楚，就能开发出相应功能的数字 IC 产品。数字 IC 产品也越来越多样化，近年来甚至出现了数字 IC 大规模替代模拟 IC 的趋势，越来越多的模拟 IC 开始采用混合设计的方式，嵌入数字设计模块，用于降低功耗，提高信号处理和存储能力。

## 1.2 现代数字 IC 设计方法的发展

在 CAD (Computer Aided Design, 计算机辅助设计) 软件工具成熟以前, 设计数字集成电路只能基于最原始的手段。工程师把电路图画在图纸上, 然后由人工完成繁重的计算验证, 工程师按照原理图手工画成多层版图 (最初的辅助工具是尺和笔), 再把版图刻成模板, 最后用模板进行集成电路工艺制造。由于各个环节都是手工设计, 出错概率很大, 严重影响了开发周期和成本。

### 1.2.1 自底向上的设计方法

自底向上 (bottom-up) 的设计方法是集成电路和 PCB (Printed Circuit Board) 的传统设计方法, 该方法盛行于 20 世纪 70 年代。自底向上的设计方法的思想是先设计底层的门电路, 再由门电路搭建模块级电路, 然后再由模块级电路搭建系统级电路。自底向上的设计流程如图 1.10 所示。

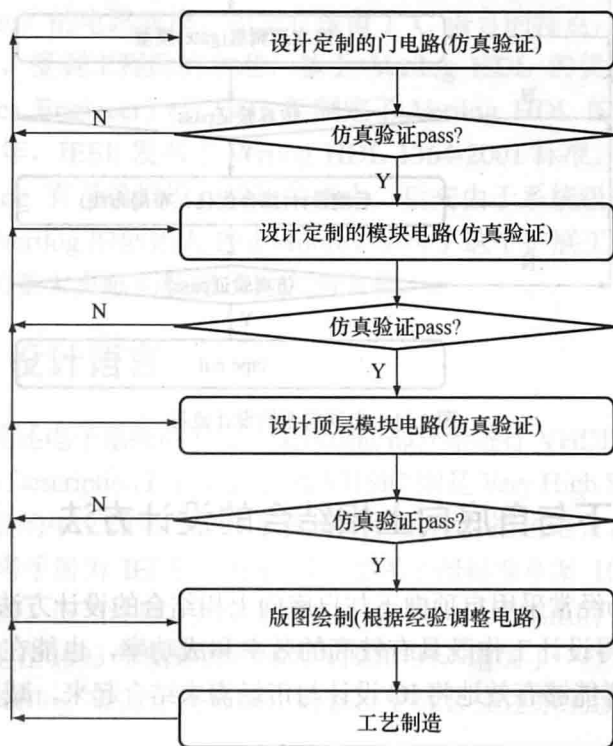


图 1.10 自底向上的设计流程

自底向上设计方法的优点是在设计早期能够准确估算性能、规模, 缺点是设计效率低、设计周期长、一次设计成功率低。

### 1.2.2 自顶向下的设计方法

自顶向下 (top-down) 设计方法的思路是: 从确定电路系统的功能和性能指标开始, 自系统级、寄存器传输级、逻辑级直到物理级, 逐级细化并逐级验证其功能和性能。具体流程如图 1.11 所示。

自顶向下设计方法的优点是设计效率高、设计周期短、一次设计成功率高，缺点是在设计早期无法准确估算电路的性能和规模等指标。

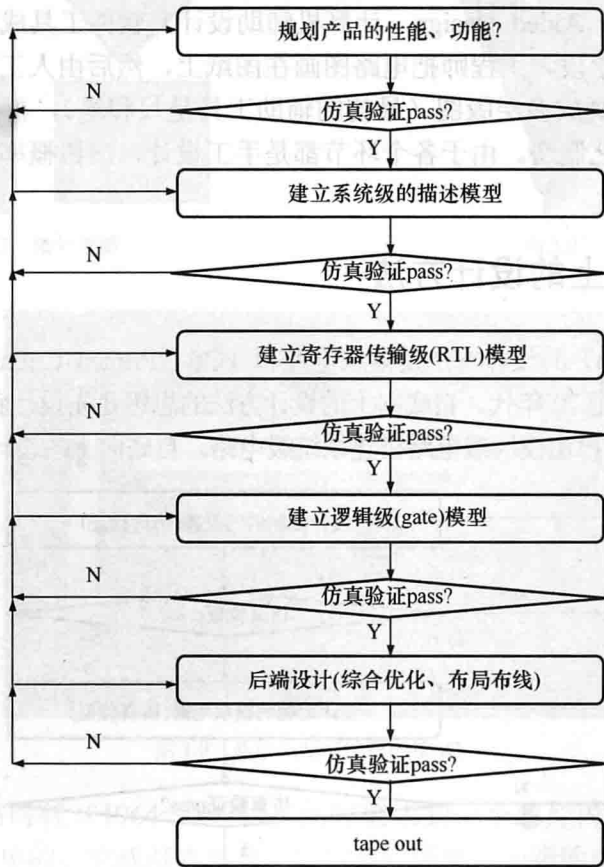


图 1.11 自顶向下的设计流程

### 1.2.3 自顶向下与自底向上相结合的设计方法

目前 IC 设计工程师经常采用自顶向下与自底向上相结合的设计方法，这样可以有效整合两种设计方法的优点，使得设计工作既具有较高的效率和成功率，也能在设计早期较准确地估算电路的性能和规模。这样能够有效地将 IC 设计与市场需求结合起来，减少产品的市场风险。

## 1.3 数字 IC 前端设计语言及后端设计软件

随着设计复杂性的提高，传统的电路原理图设计方法的工作量越来越大，验证也越来越困难，设计效率和正确性大大降低，无法满足市场对 IC 产品高性能、强功能、低成本、快上市的要求。为了解决上述问题，人们发明了硬件描述语言（Hardware Description Language, HDL），用硬件描述语言描述电路的结构和行为，建立集成电路模型并进行仿真验证，从而大大提高了集成电路设计的抽象层次，降低了设计复杂度，提高了设计效率和成功率。

硬件描述语言具有下述特点。

(1) 采用高级程序设计语言的结构，但与一般软件程序设计语言有差别，它是针对硬件



系统与部件设计的。

(2) 能够在不同的抽象层次上描述电路的功能和连接关系, 可以描述电路的结构。

(3) 除了描述功能之外还能描述时序关系。

(4) 能够描述电路固有的并行性, 可同时执行多个任务。

(5) 具有描述延迟等时间概念的能力。

目前 IC 设计领域应用最为广泛的两种硬件描述语言是 Verilog 和 VHDL。

### 1.3.1 Verilog 硬件描述语言

Verilog 硬件描述语言是由民间发展起来的, 其语法风格类似于 C 语言。1985 年, Gateway Design Automation 公司为其仿真器产品 Verilog-XL 开发了 Verilog 语言。1989 年, Cadence 收购 Gateway 公司, 并继续推广该语言和仿真器。

Verilog 硬件描述语言在 C 语言的基础上, 引入了并行计算的概念, 可以准确地描述二值电压的数字电路行为, 还可以实现系统级 (system)、算法级 (algorithm)、行为级 (behaviour)、门级 (gate) 的电路建模。由于它保留了 C 语言的特点, 简单灵活, 很快就被应用于数字电路的设计中, 受到工程师的欢迎。基于 Verilog HDL 的优越性, IEEE (Institute of Electrical and Electronics Engineer) 于 1995 年制定了 Verilog HDL 的 IEEE 标准, 即 Verilog HDL 1364-1995; 2001 年, IEEE 发布了 Verilog HDL 1364-2001 标准。在这个标准中, 加入了 HDL-A 标准, 使 Verilog 有了模拟设计描述的能力。后来由于系统级的设计需要, 由 Verilog 语言衍生出 Superlog, Verilog 的创始人 Phil Moorby 参与了这个扩展工作。Phil Moorby 本人因对现代 IC 设计自动化的重大贡献而获得了巨大的荣誉。

### 1.3.2 VHDL 设计语言

因为美国军事需要描述电子系统的方法, 美国国防部开始进行 VHDL 的开发。VHDL 的英文全名为 VHSIC Hardware Description Language, 而 VHSIC 则是 Very High Speed Integrated Circuit 的缩写, 故 VHDL 其准确的中文译名为甚高速集成电路的硬件描述语言。1987 年, 由 IEEE 将 VHDL 制定为标准, 参考手册为 IEEE VHDL 语言参考手册标准草案 1076/B 版, 于 1987 年批准, 称为 IEEE 1076-1987。应当注意, 起初 VHDL 只是作为系统规范的一个标准, 而不是为设计而制定的。第二个版本是在 1993 年制定的, 称为 VHDL-93, 增加了一些新的命令和属性。

VHDL 语言相对 Verilog 来说比较繁杂、抽象, 更适合描述系统级的模型, 所以使用的范围没有 Verilog 广泛。

随着数字芯片设计规模越来越大, 模块化的重复使用越来越多, 尤其是以处理器为核心的系统芯片, 软件设计和硬件设计越来越迫切需要一个统一的设计平台。增强系统级描述能力的 HDL 软件应运而生, 如 Superlog、System C、Cynlib C++、C Level 等, 它们各有长处, 可以大大提高超大规模芯片设计的效率。但是目前的这些增强语言中, 并没有一种可以成为像 Verilog 和 VHDL 一样广泛应用于数字设计的语言。

### 1.3.3 验证和验证语言

当数字 IC 设计进入超大规模时代时, 设计工程师发现, 设计的时间已经不是影响设计周



期的关键因素，影响设计周期的关键因素在于发现和修改设计中存在的问题。而一般的仿真工具 Verilog 和 VHDL 在仿真中只适合解决低层次的设计问题，更高层次的问题需要海量的测试矢量来发现，而产生这种测试矢量对于 Verilog 和 VHDL 来说实现起来非常麻烦，相关代码也更加繁杂，运算速度也受到制约。更重要的是，VHDL 和 Verilog 对于建立更抽象的算法级描述并不方便。

为了保证数字 IC 设计不偏离最初的目标，自顶向下的设计方法中引入了验证 (Verification) 的概念。验证的概念就是从数字设计开始，就要对系统的功能以及重要信号进行描述，以便对下一级的模型设计的结果进行比较，修正下一级模型设计中引入的偏差和错误。

System Verilog 作为一种验证语言工具，引入了软件设计中类似事件的概念，使其更适合描述芯片与芯片相关的系统，方便建立更完善的验证平台。它增强了算法的描述，更加方便描述抽象实现算法，同时又继承了 Verilog 对并行运算和位运算的支持，并且以时钟为基本时间单位，使其方便对重要信号进行的 cycle-to-cycle 比较。

C 语言和上述提到的几种语言，都有助于更加方便地建立验证的平台。

### 1.3.4 数字 IC 设计后端 EDA 工具

#### 1. 数字 IC 设计的 EDA 综合软件

EDA (Electronic Design Automation) 综合 (synthesis) 软件是从算法到门级实现的“自动化翻译软件”。综合软件的主要功能有两个：映射和优化。

业界常用的 RTL (Register Transfer Level) 级综合工具有：Synopsys 公司的 Design Compiler (DC)、Cadence 公司的 Encounter RTL Compiler、Mentor 公司的 Leo Spectrum 等。

综合工具往往包含可测性设计软件，如 DFT Compiler 等，可为同步数字电路自动生成测试电路并自动产生测试向量 (ATPG)。

#### 2. 数字 IC 设计的自动布局布线工具

自动布局布线 EDA 软件工具能够实现自动布局和布线功能，它能大大缩短数字 IC 版图的设计周期，最大限度地减少手工布局布线所产生的错误，且能修复布局布线产生的时序问题。

常用的布局布线工具有：Cadence 公司的 SOC Encounter、Synopsys 公司的 Astro 等。

#### 3. 其他后端设计分析工具

静态时序分析 (static timing analysis) 工具：用于分析数字 IC 后端设计中的时序在真实信号负载情况下是否满足寄存器时钟的建立时间和保持时间，同时可以分析电路的驱动能力及上升下降时间是否安全和稳健。

常用的静态时序分析工具：Synopsys 公司的 Prime Time 和 Cadence 公司的 ETS。

形式验证工具：保证物理版图实现的功能与 RTL 模型的逻辑功能一致。

常用的形式验证工具：Cadence 公司的 LEC (logic equivalence check) 和 Synopsys 公司的 Formality。

针对定制版图的 EDA 验证工具，有 Mentor 公司的 Calibre 和 Cadence 公司的 Assura。它们都可以做版图的工艺规则检查 (DRC)，版图和网表的等效连线检查 (LVS)，版图的电气规则检查 (ERC)。这些工具还可以做版图参数提取，反标原理图网表，以用于最后版图可靠性