21 世纪高等学校电子信息类教材

VLSI设计

王志功　朱　恩　编著

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本书介绍了 VLSI 设计的基本方法。全书共 7 章，内容包括：VLSI 设计的一般概念、方法和基本流程；Verilog 和 VHDL 语言的基本概念和用法，逻辑仿真软件 ModelSim 介绍；可编程逻辑器件基本知识和开发环境 Quartus II 介绍；逻辑综合的一般概念和方法，逻辑综合软件 Synopsys DC 介绍；自动布局、布线基本概念及 Apollo 软件介绍；SoC 基本概念，基于平台的 SoC 开发方法及 ARM 开发平台介绍；VLSI 设计的发展方向。

本书可作为电子科学和通信与信息等学科高年级本科生和硕士生的教材，也可作为集成电路设计工程师的参考书。

图书在版编目(CIP)数据

VLSI 设计／王志功，朱恩编著. —北京：电子工业出版社，2005.1
21 世纪高等学校电子信息类教材
ISBN 7-121-00621-9

Ⅰ. V… Ⅱ. ①王…②朱… Ⅲ. 超大规模集成电路—电路设计—高等学校—教材 Ⅳ. TN470.2

中国版本图书馆 CIP 数据核字（2004）第 125279 号

责任编辑：王 颖 姚晓竞
印 刷：
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
经 销：各地新华书店
开 本：787×1 092 1/16 印张：11.5 字数：294.4 千字
印 次：2005 年 1 月第 1 次印刷
印 数：4 000 册 定价：17.80 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前　言

近几年来，我国的集成电路产业得到了迅猛的发展，对集成电路设计人才的需求也在逐年提高。高校承担着人才培养的重任，为了加快人才培养速度，需要努力创造培养环境，教材建设是这个环境的重要组成部分之一。由于国内集成电路设计起步较晚，各方面的基础都比较薄弱，在教材方面，尤其缺乏大批反映当代先进的集成电路设计思想和技术的教材。

超大规模集成电路（VLSI）设计是综合性较强的集成电路设计分支，目前，VLSI 设计采用自顶向下设计方法，体现了系统工程的科学思想。但是，在过去多年对研究生的教学过程中，我们发现，国内用于介绍 VLSI 设计的自顶向下整体技术的教材较少。所以，从 1998 年开始，本书第一作者接任有关 VLSI 分析与设计的课程教学之后，就在东南大学无线电系沈永朝教授 1987—1997 年为硕士研究生授课用的、名为《VLSI 分析与设计》的手稿基础上，补充了 VHDL 语言、VLSI 逻辑综合和在系统编程技术等集成电路前端设计的内容，编写了《VLSI 设计讲义》。讲课过程中发现：(1) 研究生大多在本科阶段没有系统学习过集成电路设计；(2) VLSI 前端设计与各类集成电路的后端设计在学习内容、设计方式和应用 CAD/EDA 工具等方面都有很大不同；(3) 研究生在一个学期内难以兼顾集成电路前端设计和 VLSI 后端设计两方面的内容。所以，从 2001 年始，我们将原来的《VLSI 设计讲义》分成两门课程。一门称之为《集成电路设计基础》(已于 2004 年 6 月由电子工业出版社出版)，侧重于集成电路设计基础知识和后端设计，继续由本书第一作者讲授。另一门内容集中在前端（或称顶层）设计，由本书第二作者增加了 VLSI 设计概述、Verilog 语言、自动布局布线和 SoC 技术简介等章节，并增加了逻辑仿真软件 ModelSim、FPGA 开发软件 Quartus II、逻辑综合软件 Synopsys DC、自动布局布线软件 Apollo II 和 ARM 的 SoC PrimeXsys 平台等内容，沿用《VLSI 设计》名称，独立开课讲授。几年来作者不断地对这门课进行体系构思、内容充实和教学检验，最终形成了现在的这本《VLSI 设计》。

本教材共分 7 章，按自顶向下设计的顺序，从最上层的行为级设计开始一直讲到最底层的自动布局、布线。第 1 章介绍了 VLSI 设计的一般概念和方法，简单讨论了 VLSI 设计的流程问题，同时也简要介绍了当代 VLSI 设计的前沿问题和发展方向。第 2 章和第 3 章介绍了 VLSI 设计所需要的语言知识，简要介绍了 Verilog 语言和 VHDL 语言，没有过多地介绍语言的细节，而是从整体出发，介绍语言的最基本、最关键的用法，使初学者有更多的精力关注 VLSI 设计的整个流程。在介绍语言的过程中，介绍了一个常用的逻辑仿真软件 ModelSim，使大家在进行简短的语言学习以后，就可立即进入逻辑模拟阶段，便于读者进一步自学和掌握语言的更复杂的内容。第 4 章介绍了可编程逻辑器件的知识和开发方法，利用可编程逻辑器件进行 VLSI 设计，是业界比较流行的辅助设计方法。这种方法有利于减少 VLSI 的设计周期。在介绍了一般的可编程器件的知识以后，本章重点介绍了一个开发环境，即 Altera 公司的 QuartusII 软件，并通过一个具体的简单实例介绍了开发软件的使用流程。可编程逻辑器件的开发环境大同小异，所以介绍了一种环境，有利于读者进一步学习和掌握其他的开发工具。第 5 章介绍了逻辑综合技术的一般概念、方法以及与硬件描述语言的关系。同时，介绍了一个常用的逻辑综合工具 Synopsys DC，并通过一个具体的例子，介绍了逻辑综合的一般流程。第 6 章介绍了自动布局、布线的基本概念、方法和常用软件工具，并通过一个简单的例子，介绍了实际自动布局、布线软件的使用方法。第 7 章介绍了 SoC 的基本概念和目前的实用开

发方法。

虽然实际应用中的 VLSI 设计千变万化，但作者认为，经过这 7 章内容的学习，初学者基本可以掌握一个 VLSI 设计的完整流程了。

本书可以作为电子科学和通信与信息等学科高年级本科生和硕士生的教材，也可作为集成电路设计工程师的参考书。

在本书的编写过程中，硕士研究生蒋道三、曾敏、朱礼安和荣瑜在软件使用和部分实例设计方面做了许多工作，乔庐峰博士也提出了一些很好的建议，在此表示衷心的感谢。同时，我们也要感谢原 Avant!公司（现 Synopsys 公司）在 EDA 软件方面给予的无偿的大力支持，特别要感谢原 Avant!公司的徐建民、杨颖、井丽萍、张丽、石荣和邓泽群等给予我们的大力支持。东南大学射频与光电集成电路研究所部分老师和研究生也为本书做出了不同程度的贡献，在此表示衷心的感谢。

本教材经过了 4 届学生教学的试用，中间经过了不断的充实和完善，但集成电路技术发展迅速，涉及的技术领域众多，一本教材很难反映技术发展前沿的变化。我们虽然尽了最大努力，仍难以达到兼顾基础又跟踪前沿和兼顾全面又突出重点的多重目标。由于篇幅和我们的水平所限，目前的这本教材还不是非常理想的，但为解教学上的燃眉之急，我们还是决定出版这本书，起一个抛砖引玉的作用，希望对国内的 VLSI 设计的教学尽一点微薄之力。本书的编写时间是我们在科研和教学的过程中抽空完成的，错误之处在所难免，希望读者提出宝贵意见和建议，使本书得到完善和提高。

编著者

目　录

第 1 章　VLSI 概述 …… (1)
1.1　发展概貌 …… (1)
1.2　主要设计方法——自顶向下方法 …… (1)
1.3　VLSI 设计流程中的重点问题 …… (2)
1.3.1　高层综合 …… (2)
1.3.2　逻辑综合 …… (3)
1.3.3　物理综合 …… (3)
1.4　工具的支持 …… (4)
思考题 …… (5)
第 2 章　硬件描述语言 Verilog …… (6)
2.1　Verilog 语言的一般结构 …… (6)
2.1.1　模块 …… (6)
2.1.2　数据流描述方式 …… (6)
2.1.3　行为描述方式 …… (7)
2.1.4　结构化描述方式 …… (8)
2.1.5　混合描述方式 …… (8)
2.2　Verilog 语言要素 …… (9)
2.2.1　标识符、注释和语言书写的格式 …… (9)
2.2.2　系统任务和函数 …… (9)
2.2.3　编译指令 …… (9)
2.2.4　值集合 …… (10)
2.2.5　数据类型 …… (11)
2.2.6　位选择和部分选择 …… (13)
2.2.7　参数 …… (14)
2.3　表达式与操作符 …… (14)
2.3.1　算术操作符 …… (14)
2.3.2　关系操作符 …… (15)
2.3.3　相等关系操作符 …… (15)
2.3.4　逻辑操作符 …… (15)
2.3.5　按位操作符 …… (15)
2.3.6　归约操作符 …… (15)
2.3.7　移位操作符 …… (16)
2.3.8　条件操作符 …… (16)
2.3.9　连接操作符 …… (16)
2.3.10　复制操作符 …… (16)
2.4　结构描述方式 …… (17)
2.4.1　常用的内置基本门 …… (17)

2.4.2 门时延问题 …… (17)
2.4.3 门实例数组 …… (18)
2.4.4 模块和端口 …… (18)
2.4.5 模块实例语句 …… (19)
2.4.6 模块使用举例 …… (20)
2.5 数据流描述方式 …… (20)
2.5.1 连续赋值语句 …… (20)
2.5.2 举例 …… (21)
2.5.3 连线说明赋值 …… (21)
2.5.4 时延 …… (22)
2.5.5 连线时延 …… (22)
2.5.6 举例 …… (23)
2.6 行为描述方式 …… (24)
2.6.1 过程结构 …… (24)
2.6.2 时序控制 …… (25)
2.6.3 语句块 …… (27)
2.6.4 过程性赋值 …… (28)
2.6.5 if 语句 …… (30)
2.6.6 case 语句 …… (30)
2.6.7 循环语句 …… (31)
2.7 设计共享 …… (32)
2.7.1 任务 …… (32)
2.7.2 函数 …… (33)
2.7.3 系统任务和系统函数 …… (34)
2.8 HDL 仿真软件简介 …… (36)
思考题 …… (43)
第 3 章 硬件描述语言 VHDL …… (45)
3.1 VHDL 语言的基本结构 …… (45)
3.2 VHDL 的设计实体 …… (45)
3.2.1 实体说明 …… (45)
3.2.2 结构体 …… (46)
3.3 VHDL 中的对象和数据类型 …… (48)
3.3.1 数的类型和它的字面值 …… (49)
3.3.2 数据类型 …… (49)
3.3.3 对象的说明 …… (50)
3.3.4 VHDL 中数的运算 …… (51)
3.4 行为描述 …… (52)
3.4.1 对象的赋值 …… (52)
3.4.2 并发进程 …… (52)
3.4.3 并行信号赋值语句 …… (52)

3.4.4 进程语句 ……(54)
3.4.5 顺序赋值语句 ……(55)
3.4.6 顺序控制 ……(56)
3.4.7 断言语句 ……(58)
3.4.8 子程序 ……(60)
3.5 结构描述 ……(61)
3.5.1 元件和例元 ……(61)
3.5.2 规则结构 ……(62)
3.5.3 参数化设计 ……(63)
3.5.4 结构与行为混合描述 ……(64)
3.6 设计共享 ……(65)
3.6.1 程序包 ……(65)
3.6.2 库 ……(66)
3.6.3 元件配置 ……(67)
思考题 ……(70)
第 4 章 可编程逻辑器件 ……(71)
4.1 引言 ……(71)
4.2 GA 概述 ……(71)
4.3 PLD 概述 ……(72)
4.3.1 PLD 的基本结构 ……(72)
4.3.2 PLD 的分类 ……(72)
4.3.3 PROM 阵列结构 ……(73)
4.3.4 PLA 阵列结构 ……(73)
4.3.5 PAL(GAL)阵列结构 ……(74)
4.3.6 FPGA(Field Programmable Gate Array) ……(75)
4.3.7 PLD 的开发 ……(79)
4.4 FPGA 的开发实例 ……(80)
4.4.1 Quartus II 的启动 ……(81)
4.4.2 建立新设计项目 ……(81)
4.4.3 建立新的 Verilog HDL 文件 ……(83)
4.4.4 建立新的原理图文件 ……(84)
4.4.5 设置时间约束条件 ……(85)
4.4.6 引脚绑定 ……(86)
4.4.7 编译 ……(88)
4.4.8 仿真 ……(89)
4.4.9 器件编程 ……(91)
思考题 ……(92)
第 5 章 逻辑综合 ……(94)
5.1 引言 ……(94)
5.2 组合逻辑综合介绍 ……(94)

5.3 二元决定图（Binary-Decision Diagrams）……(96)……
5.3.1 ROBDD 的原理……(97)……
5.3.2 ROBDD 的应用……(98)……
5.4 Verilog HDL 与逻辑综合……(99)……
5.4.1 assign 结构……(100)……
5.4.2 if-else 表达式结构……(101)……
5.4.3 case 表达式结构……(101)……
5.4.4 for 循环结构……(101)……
5.4.5 always 表达式……(101)……
5.4.6 function 表达式结构……(102)……
5.5 逻辑综合的流程……(102)……
5.5.1 RTL 描述……(102)……
5.5.2 翻译……(102)……
5.5.3 逻辑优化……(103)……
5.5.4 工艺映射和优化……(103)……
5.5.5 工艺库……(103)……
5.5.6 设计约束条件……(103)……
5.5.7 最优化的门级描述……(103)……
5.6 门级网表的验证……(106)……
5.6.1 功能验证……(106)……
5.6.2 时序验证……(107)……
5.7 逻辑综合对电路设计的影响……(107)……
5.7.1 Verilog 编程风格……(107)……
5.7.2 设计分割……(109)……
5.7.3 设计约束条件的设定……(111)……
5.8 时序电路综合举例……(111)……
5.9 Synopsys 逻辑综合工具简介……(116)……
5.9.1 实例电路 —— *m* 序列产生器……(117)……
5.9.2 利用 Synopsys 的 Design Compiler 进行综合的基本过程……(118)……
思考题……(122)……
第 6 章 自动布局、布线……(123)……
6.1 自动布局、布线的一般方法和流程……(123)……
6.1.1 数据准备和输入……(123)……
6.1.2 布局规划、预布线、布局……(124)……
6.1.3 时钟树综合……(125)……
6.1.4 布线……(127)……
6.1.5 设计规则检查和一致性检查……(127)……
6.1.6 输出结果……(127)……
6.1.7 其他考虑……(127)……
6.2 自动布局、布线软件介绍……(127)……

6.2.1 Apollo 一般情况介绍 (128)
6.2.2 Apollo 库的文件结构 (128)
6.2.3 逻辑单元库——TSMC 0.25μm CMOS 库 (129)
6.3 自动布局、布线的处理实例 (129)
6.3.1 电路实例 (129)
6.3.2 数据准备和导入 (137)
6.3.3 数据导入步骤 (137)
6.3.4 布图 (139)
6.3.5 预布线 (142)
6.3.6 单元布局 (145)
6.3.7 布线 (146)
6.3.8 数据输出 (149)
6.3.9 自动布局、布线的优化 (150)
思考题 (152)
第 7 章 SoC 技术简介 (153)
7.1 SoC 的基本概念 (153)
7.1.1 SoC 的特征和条件 (153)
7.1.2 SoC 的设计方法学问题 (153)
7.2 基于平台的 SoC 设计方法 (157)
7.2.1 一般方法 (157)
7.2.2 设计分工 (159)
7.3 ARM PrimeXsys 平台 SoC 设计方法 (161)
7.3.1 简介 (161)
7.3.2 标准的 SoC 平台 (162)
7.3.3 支持工具和验证方法 (164)
7.3.4 操作系统端口 (168)
7.3.5 ARM 的扩展 IP (169)
7.3.6 第三方伙伴计划 (169)
7.4 待解决的几个研究方向 (170)
思考题 (170)
主要参考文献 (171)

第 1 章　VLSI概述

本章简单介绍 VLSI 设计的概念、方法、问题和发展趋势。

1.1　发 展 概 貌

集成电路技术发展到 20 世纪 80 年代，出现了超大规模集成电路（VLSI），人们采用计算机辅助设计（CAD）和辅助工程（CAE）技术进行芯片的设计和生产；到了 20 世纪 90 年代，出现了特大规模集成电路（ULSI），采用电子设计自动化（EDA）技术进行芯片的设计；到了 21 世纪初，半导体工艺和电路设计水平的发展已使在一个芯片上集成一个系统成为可能，即出现了 SoC（System on Chip）技术，相应地也出现了许多急需解决的技术难题。

业界许多专家估计，著名的摩尔定律还将使用至少 10 年，目前的工艺仍有巨大的改进潜力，作为 VLSI 重要基础的 CMOS 工艺特征尺寸发展到了 0.1μm 以下，就将进入纳米时代，纳米技术将为 SoC 技术的实现提供更加坚实的基础，所以 SoC 技术有着非常诱人的发展前景！

图 1.1 是业界对集成电路技术发展的一些趋势分析。

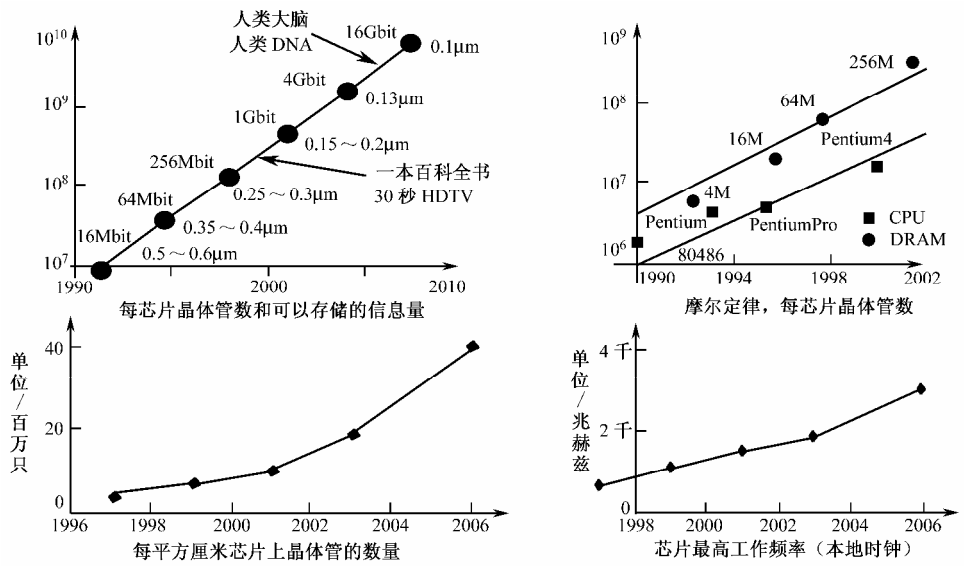

图 1.1　业界对集成电路发展趋势的分析和预测

1.2　主要设计方法——自顶向下方法

VLSI 芯片包括 SoC，基本上都采用自顶向下（Top-down）方法进行设计。

Top-down 方法的设计次序是：行为设计、结构设计、逻辑设计、电路设计和版图设计，如图 1.2 所示。

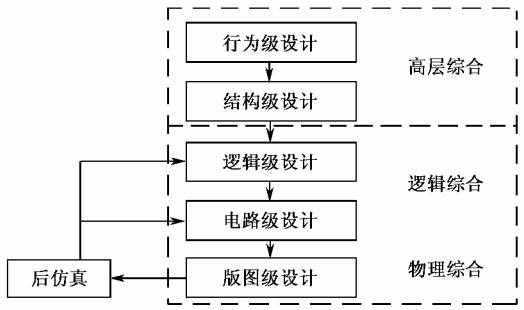

图 1.2　Top-down **方法的设计次序**

在行为级设计阶段，要确定芯片的功能、性能、面积、工艺和成本。

在结构级设计阶段，根据芯片的特点，将其分解为接口清晰、相互关系明确的、尽可能简单的子系统，再利用子系统设计出一个较好的总体结构，这个结构可能包括有算术运算单元、控制单元、数据通道和各种算法状态机等。

在逻辑级设计阶段，要考虑各种功能模块的具体实现问题。由于同一功能块可能有多种实现方法，所以在这一阶段，要尽可能采用规则结构来实现功能块，同时要充分利用已经过考验的逻辑单元或模块。这一阶段要进行逻辑仿真，以确定逻辑设计的正确性。

在电路级设计阶段，逻辑图将进一步转换成电路图，还可能需要进行电路仿真，以确定电路特性、功耗和时延等。

在版图级设计阶段，要根据电路图绘制用于工艺制造的电路版图。完成版图后，需要对实际版图进行参数提取，再进行电路后仿真。如果仿真性能达不到要求，需要返回到电路设计或更进一步到逻辑设计，进行适当修改，以最终达到设计目标。

1.3　VLSI 设计流程中的重点问题

在图 1.2 **中，典型的设计流程被划分成三个综合阶段：高层综合、逻辑综合和物理综合。**

在英语文献中，表现综合的词是 Synthesis，Synthesis **的原义是合成、综合，意指用合成方法合成出新的事物，在集成电路设计领域，综合是指将一种形式的设计转换成另一种设计形式的过程。**

1.3.1　高层综合

高层综合也称行为级综合（behavioral synthesis**），它是将系统的行为、各个组成部分的功能及其输入和输出，用硬件描述语言** HDL **（如** VHDL **和** Verilog**）加以描述，然后进行行为级综合，同时通过高层次硬件仿真进行验证。**

高层综合的任务是将一个设计的行为级描述转换成寄存器传输级的结构描述。它首先翻译和分析用 HDL **语言描述的设计，并在给定的一组性能、面积或功耗的约束条件下，确定需要哪些硬件资源，如执行单元、存储器、控制器和总线等，以及确定在这一结构中各种操作的次序。还可通过行为级和寄存器传输级硬件仿真进行验证。由于同一个设计可能有多种硬件结构的实现方式，因此，高层综合的目的就是要在满足目标和约束的条件下，找到一个代价最小的硬件结构，并使设计的功能最佳化。**

1.3.2 逻辑综合

逻辑综合将逻辑级行为描述转换成使用门级单元的结构描述（门级结构描述称为网表描述），同时还要进行门级逻辑仿真和测试综合。

逻辑级行为描述可以是状态转换图或有限状态机，也可以是布尔方程、真值表或硬件描述语言。逻辑综合过程还包括一系列优化步骤，如资源共享、连接优化和时钟分配等。优化的目标是面积、速度和功耗等指标中的一种或几种的折中。

逻辑综合一般分为两个阶段：①与工艺无关的阶段，这时采用布尔操作或代数操作技术来优化逻辑；②工艺映射阶段，根据电路性质（如组合型或时序型）及采用的结构（多层逻辑、PLD 或 FPGA）做出具体的映像，将与工艺无关的描述转换成门级网表、PLD 或 FPGA 中的一种执行文件。

在逻辑综合过程中，需要进行细致的时序和时延分析及逻辑仿真。逻辑仿真是保证设计正确的关键步骤，过去通常采用软件模拟的方法，近年来则强调硬件手段，如通过 PLD 或 FPGA 进行。测试综合是提供测试图形的自动生成，为可测性设计提供高故障覆盖率的测试图形。测试综合可消除设计中的冗余逻辑，诊断不可测的逻辑结构，还能自动插入可测性结构。

1.3.3 物理综合

物理综合也称版图综合（layout synthesis），物理综合将网表描述转换成版图。这时对每个单元确定其几何形状、大小及位置，确定单元间的连接关系。它的任务是将门级网表自动转换成版图。布图的详细步骤如图 1.3 所示。

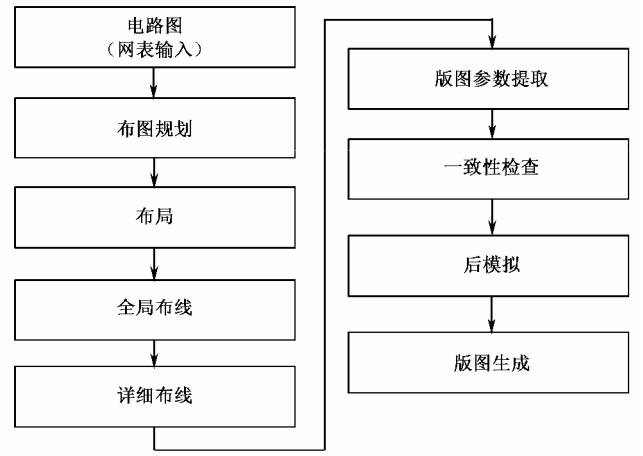

图 1.3　布图的详细步骤

布图规划是对设计进行物理划分，同时对设计的布局进行规划和分析。这一步是面向物理的划分，其层次结构可以与逻辑设计时的划分有所不同。布图规划可以估算出较为精确的互连延迟信息，预算芯片的面积，分析布线的稀疏度。

布局是指将模块在满足一定的目标函数的前提下布置在芯片上的适当位置。一般布局时总是要求芯片面积最小、连线总长最短、电性能最优并且容易布线。布局又分为初始布局和迭代改善两个子步骤。进行初始布局的目的是为了提高布局的质量及减少下一阶段即迭代改

善时的处理次数，而迭代改善是一个优化的过程，它是决定布局质量的关键。

布线是根据电路连接关系的描述（即连接表），在满足工艺规则和电学性能的要求下，在指定的区域（面积、形状、层次等）内完成所需的全部互连，同时尽可能地对连线长度和通孔数目进行优化。一般有两种布线方法；一种是面向线网的布线方法，它是直接对整个电路进行布线，布线时通常采取顺序方式；另一种称为分级布线，它将布线问题分为全局布线和详细布线，这是一种面向布线区域的布线方法。这种方法通过适当的划分，将整个布线区域分为若干个布线通道区，然后进行适当的布线分配，即将一个线网的所有端点的走线路径分配到相应的通道区中，接着是进行详细布线，对分配到当前通道区中的所有线网段的集合，按照一定的规则，确定它们在通道区中的具体位置。

在完成布局、布线后，要对版图进行设计规则检查、电学规则检查及版图与电路图的一致性检查。在版图寄生参数提取的基础上，再次进行电路分析（即后模拟）。只有在所有的检查都通过并证明正确无误后，才可将布图结果转换为掩膜文件。然后通过掩膜版发生器或电子束制版系统，将掩膜文件转换生成掩膜版。

上述设计流程又可分成前端设计和后端设计两个阶段，这是在深亚微米电路出现之前，人们就已习惯的分法。

前端设计主要进行系统和功能的设计及结构和电路的设计。

后端设计主要完成版图设计。

前端和后端之间的沟通主要通过网表和单元库，前端设计完成后将网表交给版图设计人员。过去，一般而言，只要布线能够通过，时序要求基本上能够得到满足。

1.4 工具的支持

VLSI 设计离不开工具的支持。20 世纪 70 年代初，为了解决电路和版图设计自动化问题，出现了计算机辅助设计（CAD）技术。到了 20 世纪 80 年代，为了解决结构和逻辑设计的自动化问题，出现了计算机辅助工程（CAE）技术。到了 20 世纪 90 年代初，为了解决行为设计的自动化问题，出现了电子设计自动化（EDA）技术。在 CAD/CAE 技术中，主要解决给定电路图到版图和测试生成的自动转化问题。在 EDA 技术中，主要解决给定概念到电路图的自动转化问题。新出现的技术一般涵盖和改进了已有的技术，例如 EDA 技术涵盖了 CAD/CAE 技术等。

CAD 技术的出现主要用来帮助设计人员进行电路性能的分析和完成电路版图的编辑，如何在计算机上完成复杂的运算并解决图形的编辑及设计规则的检查是需要解决的主要问题。CAD 技术主要以晶体管级的性能仿真和多层版图的手工编辑为主，并形成了众多的单点工具。CAD 技术的出现将部分设计工作从手工劳动当中解放出来，同时建立了模拟的概念和基本理论。

CAE 技术主要用来进行集成电路设计、开发与生产过程的标准化，同时实现各个层面上集成电路设计活动的自动化。这种技术需要解决的主要难题是如何建立集成电路的标准设计库，如何解决不同层面设计的接口等问题。这个时期，出现了标准设计库，出现了硬件描述语言 Verilog 和 VHDL，出现了全线设计工具。CAE 技术孕育了众多的新思想，例如可测性设计，出现了集成电路设计工具的全线产品，形成了集成电路设计、开发和生产的流程。

EDA 技术的出现主要用来实现从概念到电路的设计自动化，实现电子系统设计与芯片设

计的融合，需要解决的主要难题是电路综合理论的创立和完善，电路验证理论的出现和发展。这个时期的主要特征是广泛使用硬件描述语言和逻辑综合进行电路设计，电路验证理论和方法逐渐成熟，行为综合理论开始形成，产生了众多的新思想和新方法，如时序验证等，实现了集成电路设计工作的重点转移，形成了新一代集成电路设计方法学。

思考题

1．了解目前业界所使用的 EDA 工具的情况，针对一两个具体的集成电路 EDA 软件功能，分析它们在 VLSI 设计流程中的作用。

2．了解目前市场上最新的 CPU、DSP、通信芯片、图像处理芯片的主要功能结构、工艺、晶体管数，试给出它们的研发流程和所需的技术支持。

第 2 章　硬件描述语言Verilog

Verilog HDL 是硬件描述语言的一种，可用于从算法级、门级、寄存器级到开关级的多层次的数字电路系统建模，也可用于时序建模，用 Verilog 编写的模型可用 Verilog 仿真器进行验证。

Verilog 从 C 语言中继承了多种操作符和结构，并提供了扩展的建模能力。

Verilog 语言的国际标准为 IEEE Std 1364—1995（1995 年）。

2.1　Verilog 语言的一般结构

2.1.1　模块

模块是 Verilog 的基本描述单位，用于描述某个设计的功能或结构，同时也可描述与其他模块通信的外部端口。一个模块可以调用另一个模块。

下面是一个半加器电路的设计实例：

```
module HalfAdder (A, B, Sum, Carry);
    input A, B;
    output Sum, Carry;
        assign #2 Sum = A ^ B;
        assign #5 Carry = A & B;
endmodule
```

该模块的名字是 HalfAdder，模块的输入端口为 A 和 B，输出端口为 Sum 和 Carry。

在模块中，可用数据流描述方式、行为描述方式、结构化描述方式、混合描述方式来描述一个设计。

下面通过实例介绍这些方式。

2.1.2　数据流描述方式

用数据流描述方式描述一个设计的最基本机制就是使用连续赋值语句，例如：

```
module HalfAdder (A, B, Sum, Carry);
    input A, B;
    output Sum, Carry;
        assign #2 Sum = A ^ B;
        assign #5 Carry = A & B;
endmodule
```

模块 HalfAdder 有两个输入端口和两个输出端口，以 assign 为前缀的语句是连续赋值语句；连续赋值语句是并发执行的，各语句的执行顺序与其在描述中出现的顺序无关。

2.1.3 行为描述方式

电路的行为使用下述过程语句描述：

① initial **语句：此语句只执行一次。**

② always **语句：此语句总是循环执行。**

在这两种语句中被赋值的对象只能是寄存器类型的。

语句 initial **和** always **在** 0 **时刻开始并发执行。**

例如：

```
module FA (A, B, Cin, Sum, Cout);
    input A, B, Cin;
    output Sum, Cout;
    reg Sum, Cout;
    reg T1, T2, T3;
        always @ ( A or B or Cin )
            begin
                Sum = (A ^ B) ^ Cin;
                T1 = A & Cin;
                T2 = B & Cin;
                T3 = A & B;
                Cout = (T1| T2) | T3;
            end
endmodule
```

其中，reg **是寄存器数据类型的一种。**always **语句中字符@ 后面的是事件控制语句。相关联的顺序过程是** begin **和** end **包含的语句。只要** A、B **或** Cin **中至少有一个值发生变化，顺序过程就执行。**

下面是 initial **语句的示例：**

```
'timescale 1ns / 1ns
module Test (P);
    output P;
    reg P;
        initial
        begin
            P = 0;
            P = #5 1;
        end
endmodule
```

语句 initial **包含一个由** begin **和** end **包含的顺序过程，这一顺序过程在** 0ns **时开始执行，在语句“** P= #5 1; **”执行后，**initial **语句被永远地挂起。**

2.1.4 结构化描述方式

结构化描述方式可使用下面的结构：

① 开关级原语（晶体管级）；

② 内置门原语或用户定义的原语（门级）；

③ 模块实例（创建层次结构）。

通过使用连线来相互连接。

下面是使用内置门原语描述的全加器电路实例：

```
module FA_S (A, B, Cin, Sum, Cout);
    input A, B, Cin;
    output Sum, Cout;
    wire S1, T1, T2, T3;
        xor   X1 (S1, A, B), X2 (Sum, S1, Cin);
        and   A1 (T3, A, B), A2 (T2, B, Cin), A3 (T1, A, Cin),
        or    O1 (Cout, T1, T2, T3);
endmodule
```

在这一实例中，包含了内置门 xor、and 和 or 的实例语句。门实例由连线类型变量 S1、T1、T2 和 T3 互连。

2.1.5 混合描述方式

模块描述中可以包含实例化的门、模块实例化语句、连续赋值语句以及 always 语句和 initial 语句的混合。它们之间可以相互包含。

下面是采用混合方式设计的 1 位全加器的实例：

```
module FA_M (A ,B ,Cin ,Sum ,Cout);
    input A ,B ,Cin;
    output Sum ,Cout;
    reg Cout ,T1 ,T2 ,T3;
    wire S1;
        xor X1(S1, A, B);                  // 门实例语句
        always @ ( A or B or Cin )         // always 语句
            begin
                T1 = A & Cin;
                T2 = B & Cin;
                T3 = A & B;
                Cout = (T1| T2) | T3;
            end
        assign Sum = S1 ^ Cin;             // 连续赋值语句
endmodule
```

只要输入 A 或 B 发生变化，门实例语句即被执行；只要 A、B 或 Cin 发生变化，就执行 always 语句；只要 S1 或 Cin 发生变化，就执行连续赋值语句。