

Verilog HDL 实用教程

张明
编著



EDA 技术丛书



电子科技大学出版社

EDA 技术丛书

VerilogHDL 实用教程

张 明 编 著

电子科技大学出版社

内 容 提 要

本教材介绍在专用集成电路(ASIC: Application Specified Integrated Circuit)设计领域应用最广的硬件描述语言 Verilog HDL。掌握这种语言后,可以像编制一个软件程序一样对一个电子系统的结构或功能进行描述,通过功能仿真、逻辑综合、版图综合、时序验证等一系列后续工作,借助电子设计自动化 EDA (Electronic Design Automation) 工具,最终完成电子系统的硬件实现(IC 芯片或 FPGA 等)。

本教材主要面向有一定逻辑设计基础与程序设计经验的电子类高年级本科生或研究生,同时也是一本电子类设计从业人员必备的工程参考书。

声 明

本书无四川省版权防盗标识,不得销售;版权所有,违者必究,举报有奖,举报电话:
(028) 6636481 6241146 3201496

EDA 技术丛书

Verilog HDL 实用教程

张 明 编 著

出 版: 电子科技大学出版社(成都建设北路二段四号 邮编: 610054)

责任编辑: 张琴

发 行: 新华书店经销

印 刷: 西南冶金地质印刷厂

开 本: 797×1092 1/16 印张: 14.25 字数: 340 千字

版 次: 1999 年 11 月第一版

印 次: 1999 年 11 月第一次印刷

书 号: ISBN 7—81065—289—3 / TP · 171

印 数: 1—5000 册

定 价: 18.00 元

编 著 者 简 介

张明，1962 年 12 月生，1984 年毕业于浙江大学无线电系半导体专业，1987 年在电子部五十五所获硕士学位，1994 年在浙江大学信息与电子工程学系获通信与电子系统专业博士学位。1996 年至 1997 年在日本东京大学大规模集成电路设计与教学中心作访问研究。现为浙江大学信息与电子工程学系副教授。多年从事集成电路的科研与教学工作，1995 年完成当时国内规模最大的 16 万门图像处理专用芯片 BAP-128 的设计，并在法国 ES2 一次性流片成功。熟悉 CADENCE、SYNOPSYS 以及 COMPASS 等 EDA 工具软件的使用，并为本校研究生与本科生开设 HDL 编程与 EDA 设计相关课程。

前言

随着电子技术的不断发展与进步,传统的手工设计过程已经被先进的自动化设计工具(从CAD到EDA及ESDA: Electronic System Design Automation)所替代。因而熟悉并掌握这些现代设计工具,已成为电子系统设计人员所必备的一门技术。

通常把电子系统设计人员分为硬件和软件两类,而正是由于硬件描述语言(HDL: Hardware Description Language)的出现,打破了硬件设计与软件设计之间的界限。现代的EDA/ESDA设计工具,无不例外地以HDL作为其设计的输入,掌握EDA/ESDA工具,首先必须学会HDL编程。

提到HDL语言,大多数读者可能马上就会联想到VHDL语言。的确,它是国际上第一个标准化的HDL语言(IEEE-1076),VHDL产生的最初动因,就是为了实现美国国防部VHSIC(Very High Speed Integrated Circuit)计划所提出的各个电子部件供应商具有统一数据交换格式的要求。VHDL就是VHSICHDL的简称。相比之下,VerilogHDL则是在全球最大的EDA/ESDA供应商CADENCE公司的扶植下针对EDA工具开发的HDL语言。两者相比:(1)从推出过程可见,VHDL偏重于标准化的考虑,而VerilogHDL与EDA工具的结合则更为密切。(2)与VHDL相比,VerilogHDL的编程风格更加简洁明了,高效便捷。如果单纯从描述结构上考察,两者的代码之比为3:1。(3)VerilogHDL也已于1995年实现标准化(IEEE-1364)。目前市场上所有EDA/ESDA工具都同时支持这两种语言,而在ASIC(Application Specific Integrated Circuit)设计领域,VerilogHDL占有明显的优势。然而,面对这样的一个客观现实,国内在出版了多本介绍VHDL书籍的同时,却很难找到合适的介绍Verilog的教材。因而尽早出版一本高质量VerilogHDL教材已是当务之急。

本教材的编排如下,第一章作为全书的引论,着重介绍了与掌握HDL密切相关的EDA领域的最新发展。第二章是VerilogHDL设计入门篇,目的是让大家可以很快地对VerilogHDL的设计过程有个大致的了解。第三章至第八章具体介绍VerilogHDL各部分的语法规则。第九章是一些典型的编程错误分析与实例介绍。第十章是VerilogHDL设计的高级篇,介绍与逻辑综合相关的编程风格(Coding Style)。最后的附录A给出了VerilogHDL完整的BNF语法描述。附录B列出了一些参考文献。

本教材的使用,主要面向具有一定的逻辑电路设计基础及程序设计经验的电子类高年级本科生或研究生,编者强烈建议将HDL的内容安排在本科生的学习阶段中去,并尽快使之得到普及,让他们能够有机会及早熟悉与掌握现代EDA设计技术。本教材也可以作为电子类设计从业人员的工程参考书。对于已有一定基础的读者可以先看第一章和第二章,了解EDA领域的最新发展及用VerilogHDL进行硬件设计的Top-down实现过程,之后可直接阅读第九章和第十章,第三章至第八章的内容则可以依据自身的熟悉程度,选择性地阅读。相信对于熟悉C语言编程的读者来说,会很快喜爱并习惯用VerilogHDL进行硬件系统的设计。

编者自90年代初接触EDA工具进行集成电路设计,1995年用COMPASS成功设计了

当时国内规模最大的 16 万门图像处理专用集成芯片 BAP-128, 该芯片在法国 ES2 一次性流片成功, 并顺利地完成了国防科工委八·五攻关项目。1996~1997 年在日本东京大学大规模集成电路设计与教学中心作访问研究期间, 基本掌握了 CADENCE 和 SYNOPSYS 软件的使用, 1997 年底返回浙大后着手筹建 ASIC 设计研究室。在“211”工程经费的支持下, 研究室已配备了业界领先的 SYNOPSYS 逻辑综合软件, 之后又在国家计委数字化产业项目经费的支持下, 购买了 CADENCE EDA / ESDA 工具。正是在学校及所在系的各级领导的重视和关怀下, 使得研究室具备了一流的软、硬件设备条件, 为今后科研和教学工作的开展打下了坚实的基础。在这里我要特别感谢我的恩师姚庆栋教授, 感谢他多年来对我本人的培养以及在支持专用集成电路设计方面所倾注的心血。

最后, 编者感谢为编写本教材提供帮助的各著名 EDA 公司在中国代表(代理)处的同仁, 他(她)们是 SYNOPSYS 公司的廖彤一小姐、林倚天先生, CADENCE SYSTEM DESIGN 公司的李叶先生、张剑飞先生, AVANT! 公司的刘卫东先生。编者还要感谢电子科技大学出版社的张琴小姐, 正是她的盛情相邀才有这本教材的面世。

编 者

1999 年 10 月于浙江大学信电系

ASIC 设计研究室

Email: zhangm@isee.zju.edu.cn

目 录

第一章 电子系统设计与硬件描述语言	(1)
1.1 电子系统设计技术的发展过程	(1)
1.1.1 手工设计阶段	(1)
1.1.2 CAD 设计技术	(1)
1.1.3 EDA 设计技术	(2)
1.1.4 SOC 与 ESDA 设计技术	(3)
1.2 EDA 设计主要流程	(3)
1.3 深亚微米技术给电子系统设计技术带来的挑战	(5)
1.3.1 半导体工艺技术的进步	(5)
1.3.2 深亚微米技术带来的挑战	(6)
1.3.3 EDA 设计技术的最新发展	(7)
1.3.4 对设计人员的素质提出更高的要求	(9)
1.4 硬件描述语言	(10)
1.4.1 用 HDL 进行电子系统设计的优点	(10)
1.4.2 Verilog HDL 的发展过程与编程特点	(12)
习题	(14)
第二章 Verilog HDL 设计入门	(15)
2.1 Bottom-up 与 Top-down 设计方法	(15)
2.1.1 Bottom-up 的设计方法	(15)
2.1.2 Top-down 的设计方法	(16)
2.2 Verilog HDL 的基本结构与硬件描述的设计过程	(18)
2.2.1 Verilog HDL 与 C 语言的比较	(18)
2.2.2 module 模块的基本结构	(18)
2.2.3 行为描述与结构描述、分级设计	(20)
2.2.4 仿真与测试	(22)
2.3 用 Verilog HDL 进行硬件设计的实现过程	(24)
习题	(25)
第三章 Verilog HDL 基础知识	(26)
3.1 基本词法定义	(26)
3.1.1 空白符与注释	(26)
3.1.2 逻辑与数字表示	(27)
3.1.3 字符串与字符变量	(29)

3.1.4 取名规则、系统命令、关键词	(30)
3.2 数据类型	(32)
3.2.1 连线类型 (Net-type)	(32)
3.2.2 寄存器类型 (Register-type)	(34)
3.2.3 标量与矢量, 标量类矢量与矢量类矢量, 存储器的表示	(37)
3.3 参数定义、宏替换及模拟时间单位的定标	(39)
3.3.1 参数定义语句 parameter	(39)
3.3.2 宏替换`define	(40)
3.3.3 模拟时间定标`timescale	(41)
3.4 运算符	(42)
3.4.1 算术运算符	(44)
3.4.2 位运算符	(45)
3.4.3 缩位运算符	(47)
3.4.4 逻辑运算符	(48)
3.4.5 关系运算符	(49)
3.4.6 相等与全等运算符	(50)
3.4.7 逻辑移位运算符	(51)
3.4.8 连接运算符	(52)
3.4.9 条件运算符	(52)
3.4.10 对运算结果位宽情况的说明	(53)
3.4.11 对实型量运算的限制	(54)
习题	(55)
第四章 Verilog 行为描述	(56)
4.1 Verilog HDL 行为描述的构成框架	(56)
4.1.1 过程块	(56)
4.1.2 过程语句 initial 与 always	(57)
4.2 块语句	(58)
4.2.1 串行块 begin-end	(58)
4.2.2 并行块 fork-join	(60)
4.2.3 有名块 Named-block	(62)
4.3 赋值语句	(62)
4.3.1 什么是过程赋值语句	(63)
4.3.2 过程赋值语句的两种延时模式	(63)
4.3.3 阻塞型过程赋值与非阻塞型过程赋值	(66)
4.3.4 连续赋值语句 assign	(70)
4.3.5 过程连续赋值语句(assign/deassign, force/release)	(72)
4.4 高级程序语句	(73)
4.4.1 if-else 条件语句	(74)

4.4.2	case 语句	(78)
4.4.3	forever 循环语句	(81)
4.4.4	repeat 循环语句	(82)
4.4.5	while 循环语句	(84)
4.4.6	for 循环语句	(85)
4.4.7	disable 循环中断控制语句	(87)
4.4.8	wait 语句与有名事件	(88)
4.5	Verilog 任务与函数	(93)
4.5.1	任务 (task)	(93)
4.5.2	函数 (function)	(95)
	习题	(98)
第五章	Verilog 系统函数与编译向导	(101)
5.1	Verilog 系统任务与系统函数	(101)
5.1.1	系统任务 \$display 与 \$write	(102)
5.1.2	系统任务 \$monitor	(106)
5.1.3	系统函数 \$time 与 \$realtime	(108)
5.1.4	系统任务 \$finish 与 \$stop	(109)
5.1.5	系统任务 \$readmem	(109)
5.1.6	系统函数 \$random	(110)
5.2	编译向导	(111)
5.2.1	文件包含 `include	(111)
5.2.2	工作库定义 `uselib	(112)
5.2.3	条件编译 `ifdef, `else, `endif	(113)
5.2.4	缺省连线类型 `default_nettype	(114)
5.2.5	取消编译向导 `resetall	(115)
	习题	(115)
第六章	Verilog 结构描述(一): 门级描述	(116)
6.1	Verilog HDL 结构描述	(116)
6.1.1	module 模块的组成	(116)
6.1.2	模块调用与端口名对应方式	(117)
6.1.3	Verilog 结构描述方法	(119)
6.1.4	阵列调用(Array Instances)的结构描述	(120)
6.1.5	分级结构与分级名(Hierarchical Name)	(121)
6.2	Verilog 基本门级元件	(123)
6.2.1	Verilog 基本门级元件的定义	(123)
6.2.2	基本门的调用	(125)
6.3	Verilog 连线类型与 assign 连续赋值语句	(127)

6.3.1 连线的类型说明	(127)
6.3.2 assign 连续赋值语句	(128)
6.4 基本门与连线的延时表示	(129)
6.4.1 延时的基本表示法	(130)
6.4.2 最小延时、典型延时、最大延时	(130)
6.5 延时说明块(Specify Block)与时序检测	(131)
6.5.1 延时说明块的定义	(131)
6.5.2 延时参数定义语句 specparam	(132)
6.5.3 路径延时的定义	(132)
6.5.4 模块的时序检测	(136)
6.6 参数化模块的调用	(137)
6.6.1 直接改写调用方式	(137)
6.6.2 参数重定义(defparam)调用方式	(138)
习题	(139)
第七章 用户自定义元件(UDP)	(142)
7.1 UDP 的基本定义	(142)
7.2 组合逻辑元件的 UDP 描述	(143)
7.3 电平触发时序元件的 UDP 描述	(146)
7.4 边沿触发时序元件的 UDP 描述	(146)
7.5 电平、边沿混合触发时序电路的 UDP 描述	(148)
7.6 时序检测告警信号在 UDP 中的应用	(150)
习题	(151)
第八章 Verilog 结构描述(二): 开关级描述	(152)
8.1 Verilog 基本开关级元件	(152)
8.2 开关级元件对信号强度的影响	(153)
8.3 开关级描述中的电荷衰减表示	(156)
8.4 开关级描述的应用	(158)
习题	(161)
第九章 Verilog 设计错例与实例分析	(162)
9.1 Verilog 设计常见错误分析	(162)
9.1.1 使用非 Verilog 表达方式	(162)
9.1.2 与类型说明相关的错误	(163)
9.1.3 与 always 过程语句相关的错误	(165)
9.1.4 由零延时引起的模拟错误	(168)
9.2 Verilog 设计实例	(169)
9.2.1 常用组合电路的设计	(169)

9.2.2 常用时序电路的设计	(173)
9.2.3 有限状态机的设计	(175)
9.2.4 存储器的描述	(180)
习题	(182)
第十章 面向综合的设计技术	(183)
10.1 语言描述与综合实现的关系	(183)
10.1.1 算法描述与硬件结构	(184)
10.1.2 算法描述与资源共享	(186)
10.1.3 过程语句与隐含寄存器	(187)
10.2 Verilog HDL 描述的可综合性分析	(190)
习题	(191)
附录 A Verilog HDL 句式定义	(192)
附录 B 参考文献	(213)

第一章 电子系统设计与硬件描述语言

【本章提要】这一章作为全书的引论，结合半导体工艺与计算机技术的发展过程，简要地介绍了现代电子系统自动化设计 EDA / ESDA 技术的发展历史，以及 HDL 硬件描述语言的产生与发展过程。围绕作为一个从事电子系统的设计人员所特别关注的话题，重点强调了现代 EDA 技术的最新发展动态及其相关新技术、新思想的出现。

1.1 电子系统设计技术的发展过程

1.1.1 手工设计阶段

传统的手工设计方法就是依据所设计电子系统的实际要求，先进行功能划分，然后对每块电路画出真值表，用卡诺图进行逻辑简化，并写出布尔表达式，画出相应的逻辑线路图，再据此选择元器件，搭制线路版，最后进行实测与调试。如果是做集成电路，还必须手工绘制版图，及刻红膜等一系列复杂的手工操作。

对手工设计方法作一简单总结，其缺点突出表现在以下几个方面：

1. 复杂电路的设计、调试十分困难。
2. 如果某一过程存在错误，查找和修改十分不便。
3. 设计过程中产生大量文档，管理起来十分不便。
4. 对于集成电路设计而言，设计的实现过程与具体生产工艺相关，没有互换性。
5. 只有在设计出样机或生产出芯片后才能进行实测。

1.1.2 CAD 设计技术

自从计算机出现之后，人们可以借助于计算机进行辅助设计（CAD: Computer Aided Design），完成图形（逻辑图、版图等）输入与编辑工作，并为设计过程提供各个阶段的模拟仿真手段。对于集成电路版图设计而言，还可以进行设计过程的几何规则检查（DRC: Design Rule Check）、电学规则检查（ERC: Electrical Rule Check）。在后期还进一步实现了自动参数提取，LVS（Logic Versus Schematic）验证等一系列辅助手段。

与传统的纯手工设计相比，CAD技术的出现使电子系统设计技术向前迈出了一大步。CAD技术从70年代中后期开始产生，到80年代中后期逐渐成熟。这一阶段代表性的产品有Daisy、Tanner等。

1.1.3 EDA 设计技术

到了80年代后期, CAD技术日益成熟, 从最初单纯的作为辅助设计手段, 逐渐发展到可以提供各种自动综合工具, 如逻辑综合, 版图综合, 测试综合等, 从而真正跨入电子设计自动化(EDA: Electronic Design Automation)时代。完成从CAD到EDA的突破主要体现在以下两个方面:

一、采用硬件描述语言作为设计输入

由于HDL(HDL: Hardware Description Language)语言能够实现从抽象的行为与功能描述到具体的内部线路结构描述, 从而可以在电子设计的各个阶段、各个层次进行模拟验证, 保证设计过程的正确性。可以大大降低设计成本, 缩短设计周期。

二、库(Library)的引入

EDA工具之所以能够完成各种自动设计过程, 关键是有各类库的支持。如逻辑模拟时的模拟库、逻辑综合时的综合库、版图综合时的版图库、测试综合时的测试库等。这些库都是EDA设计公司与半导体生产厂商紧密合作, 共同开发的。

目前国际上代表性的EDA供应商有CADENCE, SYNOPSYS, AVANT!, MENTOR等, 这些公司都在我国设有代表或代理处, 考虑到国内的ASIC设计用户可能选用其中一家EDA公司的软件产品, 这里按照这些公司成立的先后顺序, 作一简要介绍。

1. MENTOR (Mentor Graphic) 公司

成立最早(1981), 从PCB起步, 涉足EDA整个设计流程, 目前在自动测试方面占有一定优势。

2. CADENCE (Cadence System Design) 公司

CADENCE的前身是成立于1982年的ECAD公司, 1988年与SDA公司合并为CADENCE公司, 提供EDA整个设计流程。目前在前端仿真(Verilog-XL)及后端布图方面占优。1997年购并HLDS公司和CCT公司, 1998年宣布购并Ambit公司。CADENCE公司自1991年以来其销售业绩一直占据EDA行业第一的位置。

3. SYNOPSYS 公司

SYNOPSYS的前身是成立于1986年的 Optimal Solution公司, 1987年原Daisy System公司的总经理Harvey Jones先生加盟任执行总裁兼总经理, 并将公司改称为SYNOPSYS。提供VHDL仿真(VSS)、逻辑综合及IP宏单元(DesignWare)开发。其逻辑综合工具Design Compiler占据80%以上的市场份额。1997年购并Viewlogic, 从而可以向客户提供Verilog HDL仿真器VCS。SYNOPSYS公司在过去的几年中得到了快速发展, 其销售业绩迅速上升为EDA行业第二的位置。

4. AVANT! 公司

AVANT!的前身是成立于1991年的ArcSys公司, 1995年11月和 Integrated Silicon Systems公司合并成立AVANT!公司, 以提供后端布图与参数提取验证工具为主, 也提供前端仿真与

形式验证工具。1997年收购COMPASS, 之后又经过多次购并, 公司得到快速成长, 成为最近几年EDA行业发展最快的公司。

EDA行业发展十分迅速, 竞争也非常激烈。总体上说, 在逻辑综合方面, SYNOPSYS独占鳌头, CADENCE则在前端仿真及后端版图设计工具上继续保持优势, 但AVANT! 已成为其强有力的竞争对手, 尤其是在超深亚微米(VDSM)设计领域, 后者的表现非常突出。MENTOR则在自动测试与提取验证工具方面尚占有一定优势。

1.1.4 SOC 与 ESDA 设计技术

进入 90 年代中期后, EDA 技术已趋向成熟, 但由于半导体工艺技术的不断进步, 使得原有的先进设计技术, 很快由于工艺水平的提高而变得落伍。在本章第三节中将专门就半导体工艺水平的发展与 EDA 设计技术的关系进行阐述。这里所强调说明的是, 在最近的几年里, 工艺上将可以实现一亿个晶体管(100M)的集成规模, 如此大规模的集成电路, 实际上已经可以把一个完整的电子系统集成到一个芯片上, 因而称之为单片系统、片上系统或系统芯片(SOC: System On a Chip)。相应要求软件公司提供系统级的设计工具, 也就是要将 EDA 提升到 ESDA (ESDA: Electronic System Design Automation 电子系统设计自动化技术)。但现有的 HDL 语言只是提供了行为级, 或功能级的描述, 尚无法完成对复杂的系统级的抽象描述。人们当然可以尝试开发一种新的系统级设计语言来完成这一工作, 但事实上, 许多电子系统设计人员已习惯于用 C 语言(或其它高级语言如 FORTRAN 等)进行系统级描述与系统级仿真, 正因如此, 更多的人把精力集中到如何将 C 语言直接转化为 EDA 系统可接受的 HDL 表述。目前这项研究已经取得了一定的进展, 但尚不能实用化。作为迈向 ESDA 的第一步, 现在已开发出系统级混合仿真工具(Co-Simulation), 代表性的产品有 SYNOPSYS 公司的 COSSAP 和 CADENCE 公司的 SPW, 它们都是针对目前发展最快的 DSP (DSP: Digital Signal Process) 及通信领域开发的系统级设计工具, 可以在同一个开发平台上, 完成高级语言(如 C 语言、FORTRAN 等)与 HDL 语言(Verilog HDL、VHDL)或其它更低层次描述模块的混合仿真。虽然用户用高级语言编写的模块尚不能自动转化成 HDL 描述, 但作为一种针对特定应用领域的开发工具, 软件供应商已经为常用的功能模块提供了丰富的宏单元库支持, 可以方便地构建应用系统, 并通过仿真加以优化, 最后自动产生 HDL 代码, 进入下一阶段的 ASIC 实现。所以客观地讲, 目前的 ESDA 技术才刚刚起步, 但可以预计在最近的几年中必将取得突破性的进展。

1.2 EDA 设计主要流程

考虑到 EDA 技术是目前较成熟的设计技术, 这一节对 EDA 设计的主要流程作一简单介绍, 读者可以从中考察 HDL 语言在其中所起的作用。

第一步: 行为级描述

在完成系统性能分析与功能划分的基础上, 对于各个电路功能模块, 用 HDL 语言(Verilog

HDL/VHDL) 完成行为级 (Behavior Level) 描述。

第二步：行为级优化与RTL级描述的转化

对上一步中完成的描述进行行为级算法优化与功能仿真，行为级算法优化的目标是选择最优的算法实现方法，行为级仿真的目的是为了验证给定的行为描述是否能够实现所需的功能。在进行行为级优化的同时，通常需要完成向RTL级描述的转化。进行这一步转化工作的原因在于，现有的EDA工具只能接受RTL级 (RTL: Register Transport Level 寄存器传送级) 描述的HDL文件进行自动逻辑综合。当然对转化后生成的RTL级描述同样需要进行仿真验证。SYNOPSIS提供的Behavior Compiler就是专门为行为级优化与RTL级描述转换而提供的单点工具。

----- 以上步骤与具体的实现工艺无关 -----

第三步：选定工艺库，确定约束条件，完成逻辑综合与逻辑优化

逻辑综合与逻辑优化 (Logic Synthesis & Logic Optimization) 的目标是将前面得到的RTL级HDL代码映射到具体的工艺上加以实现，因而从这一步开始，设计过程与实现工艺相关联。

上一节中已经提到，实现自动逻辑综合的前提是有逻辑综合库的支持，而综合库内部则包含了相应的工艺参数，最典型的如门级延时、单元面积、扇入扇出系数 (Fanin, Fanout) 等。设计一个电子系统，总有相应的设计指标，典型的如时钟频率、芯片面积、端口驱动能力等，自动综合工具将这些设计指标作为综合过程的约束条件，在给定的包含工艺参数的综合库中选取最佳单元，实现综合过程。对于比较复杂的设计，通常需要考虑测试问题 (DFT: Design For Test)，可通过测试综合工具在逻辑电路内部安排相应的测试电路。

另外需要专门强调的是，除了综合过程与工艺相关外，对应不同的实现方式还应考虑选用不同的综合工具。对于面向IC(Integrated Circuit)芯片设计的用户，SYNOPSIS的Design Compiler无疑是最成功的。CADENCE的AMBIT在综合速度、用户界面等方面，也有它独到的优势。但如果是面向FPGA设计，由于其结构的特殊性，FPGA开发商通常提供其专用的综合工具，EDA公司也提供了相应的单点工具，如SYNOPSIS的FPGA Compiler等。

第四步：门级仿真

实际上，在EDA设计过程的每一个阶段，都需进行模拟仿真，以期尽早发现并改正错误，保证设计过程的正确性。与前面的行为级仿真和RTL级仿真不同的是，完成逻辑综合后的门级仿真包含了门单元的延时信息，因而门级仿真需要相应工艺的仿真库的支持，但作为仿真器本身通常就是同一个单点工具，如CADENCE的Verilog-XL仿真器可以完成以上各个抽象层次的仿真。另外，EDA厂商通常提供专门的时序分析工具，如CADENCE的Pearl和SYNOPSIS的Prime Time 等，用以验证综合后产生的电路在时序上是否正确。

第五步：测试生成

完成逻辑综合后，可产生相应的网表文件 (Netlist)，但在将设计提交给下一步进行布

局布线时，同时应当提供相应的测试文件。

测试分为功能测试（Function Test）与制造测试（Manufacture Test）两部分。功能测试就是为了检验线路的逻辑、时序等是否正确。前面讲到EDA设计过程的每一步都需进行模拟仿真，这些都是围绕电路的功能进行的，因而都属于功能测试的范畴。制造测试则是针对半导体工艺而设计的，目的是实现高的故障覆盖率（对于FPGA设计的用户来说，不必做这方面的考虑），通常称之为测试向量。对于较复杂的时序电路而言，高故障覆盖率的测试向量必须借助于测试综合才能完成，可以自动生成测试矢量（ATPG：Automatic Test Pattern Generation）。

----- 以上设计步骤通常称之为前端设计 -----

第六步：布局布线（P&R: Place and Routing）

对于IC芯片设计来说，这一步是借助于版图综合的自动布局布线工具，在对应工艺的版图库支持下完成的，通常称之为后端设计。其中以CADENCE 和AVANT! 公司所提供的后端设计工具最为著名。对于FPGA设计来说，只需借助于FPGA供应商提供的相应P&R工具实现。

第七步：参数提取

在前面完成逻辑综合所产生的门级网表文件中，已经包含了门级单元本身的工艺参数，完成版图综合后，由于布局布线都已确定，可以从版图进一步提取出连线电阻、连线电容等分布参数。

第八步：后仿真

所谓后仿真，就是将上一步中提取的分布参数再反标到原来的门级网表中，进行包含门延、连线延时的门级仿真。这一步主要是进行时序模拟，考察在增加连线延时后，时序是否仍然满足设计要求。如果不能满足，通常需回到第三步重新确定约束条件，进行优化迭代。有时候也可能要回到第二步，从算法实现上加以调整。

第九步：制版、流片

在利用EDA工具完成设计后，就可交付半导体厂商进行投片生产。对于FPGA设计的用户只需直接写入代码即可。

1.3 深亚微米技术给电子系统设计技术带来的挑战

1.3.1 半导体工艺技术的进步

如今电子产品几乎到了“无孔不入”的地步，世界上各行各业、各个领域都有电子产品的应用需求，正是这一巨大的需求市场，推动着集成电路产业的飞速发展。作为集成电

路产业的基础,这些年来半导体工艺技术一直在得到迅速提高。集成规模按照摩尔定律每三年集成度增长四倍,最小特征尺寸缩小30%,而芯片尺寸每年提高12%。表1-1反映了近年来的这种发展趋势。其中,0.18 μm 的设计技术已在1999年提前实现。

表 1-1 半导体工艺技术的进步及相关特征参数的变化

年代	1989	1992	1995	1998	2001
集成规模 (gate)	20~100k	100~200k	200k~1M	1~2M	>4M
最小尺寸 (μm)	0.8	0.5	0.35	0.25	0.18
连线延时(ns/cm)	0.9	1.5	2.6	3.6	6.1
典型频率(MHz)	30	60	100	200	400
互连线总长(km)	0.06	0.15	0.38	0.84	2.1

1.3.2 深亚微米技术带来的挑战

当半导体工艺的最小特征尺寸小于1 μm 时,称之为亚微米设计技术,当最小特征尺寸小于0.5 μm 时,称之为深亚微米设计技术(DSM: Deep Sub Micrometer),而当进一步小于0.25 μm 时,则称之为VDSM(VDSM: Very Deep Sub Micrometer)。进入深亚微米设计后,原有的EDA设计技术受到了很大的挑战,主要表现为以下几个方面:

一、连线上的延时迅速上升,将导致原有EDA设计过程不收敛

在上一节中我们已经介绍了EDA的主要设计流程,从中可以看到,在进行逻辑综合的优化与时序分析时,考虑的只是门级单元的延时,而只有在完成版图综合后,才能得到连线上的延时分布参数,然后再把这部分的连线延时反标到门级网表中进行时序验证(后模拟)。

当特征尺寸大于0.6 μm 时,电路的延时主要集中在门级单元的时延上,如果门延占系统延时70%以上,则依据门延完成综合优化与时序仿真后,由版图综合反标回来的延时对系统时序的影响很小,后模拟通常可以一次通过。但从表1-1中可以看到,当特征尺寸进一步缩小时,单位连线上的延时及互连线的总长度迅速上升,这两方面的因素都使得连线延时在系统总延时上所占的比重越来越大。通常在0.5 μm 设计时,连线延时已经达到了总延时的50%,于是增加从版图综合反标回来的连线延时后,往往使后模拟不能一次通过,这就需要重新回到逻辑综合进行再优化设计。不难想象,进入到0.35 μm 、0.25 μm 以后,连线延时将占系统总延时的70%以上,而延时参数的分布又直接取决于版图的具体实现,按照原有的EDA流程,每进行一次逻辑优化,都需重新进行版图综合,而新的版图综合结果又产生不同的延时分布,最终将导致设计目标无法实现。因而必须引入新的设计手段,保证优化迭代过程的收敛性。