



**ELECTRONIC
ENGINEER**

XIDIAN UNIVERSITY PRESS

VHDL and Complex Digital System Design

VHDL 与复杂数字系统设计

金 西 编著

VHDL and Complex Digital System Design



西安电子科技大学出版社

<http://www.xduph.com>

VHDL 与复杂数字系统设计

金 西 编著

西安电子科技大学出版社

2003

16
9
19
2
38
20
28
45
30
39
22
22
2
10
18
20
18
22
40
22
22
20
1
1
2
1
1
2
1
1
1
1
2
2
3
1
1
2
1
1
1
1
1
1
2
2

内 容 简 介

集成电路技术迅速朝着更高集成度、超小型化、高性能、高可靠性和低功耗方向发展,EDA技术的进步是一个不可忽视的原因。本书以国内最先进的 $0.25\mu\text{m}$ 复杂数字系统设计流程为背景主线进行讲述,结合作者实践和多年积累的资料编著而成。

全书共分8章,分别介绍了复杂数字系统设计与EDA工具发展趋势,硬件描述语言VHDL,设计输入与仿真综合,EDA技术与可编程ASIC设计实现,版图编辑与版图验证,建库设计,Pspice与器件模型,测试技术与物理仿真等。本书的内容贯穿了复杂数字系统的前端设计、后端设计、建库以及可测试性设计全部流程,并以图文并茂的方式致力于设计方法的讲解。

本书可作为大专院校电子工程、计算机、微电子学与固体电子学本科生和研究生教材,也可供高新技术和信息产业的研究人员、技术人员、管理人员阅读。

图书在版编目(CIP)数据

VHDL与复杂数字系统设计/金西编著.

—西安:西安电子科技大学出版社,2003.3

ISBN 7-5606-1209-1

I. V… II. 金… III. 硬件描述语言,VHDL—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2003)第004294号

策 划 陈宇光

责任编辑 杨宗周

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8227828 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安文化彩印厂

版 次 2003年3月第1版 2003年3月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 20.875

字 数 493千字

印 数 1~4 000册

定 价 29.00元

ISBN 7-5606-1209-1/TN·0216

XDUP 1480001-1

* * * 如有印装问题可调换 * * *

前 言

随着集成电路技术的高速发展,数字系统迅速朝着更高集成度、超小型化、高性能、高可靠性和低功耗的系统级芯片(SoC, System on Chip)方向发展,一个芯片可集成高达几亿到几十亿个晶体管。之所以如此,除了工艺技术、设备和材料等因素外,计算机辅助设计技术的进步是一个不可忽视的原因。人们从实践中得到了一个毋庸置疑的结论:VLSI和SoC的进一步发展与应用离不开设计自动化(DA)和计算机辅助设计。

本书以国内最先进的 $0.25\ \mu\text{m}$ 复杂数字系统设计流程为背景主线进行叙述,结合作者实践和多年积累的资料编写而成。目的是使每一位读者立足现有条件,了解和掌握深亚微米时代复杂数字系统芯片的设计方法和所用到的EDA工具,为走上社会进入专业的电子技术公司后,能胜任各种电子产品集成化的实际设计工作打下基础。本书有关VHDL语言、L-edit、MAX+PLUS II、Foundation等内容均可在普通PC机上学习和验证,对用MAX+PLUS II实现交通灯控制器程序和用Foundation实现电压信号测量A/D应用可以作为本科生毕业设计参考和指导。本书限于篇幅,对用VHDL实现的组合逻辑、时序逻辑以及像在CDMA应用的Gold码发生器,虽只给出了典型范例,但也提供了各类典型应用有关的网站和参考资料,便于读者立足本书所给的典型范例,能更进一步学习并获得解决实际问题的能力。而对于诸如Cadence的版图、Dracula、Hspice等大型昂贵软件,更注重设计入门方法和流程的讲解。

中国科技大学物理系教学办公室李永平教授、魏海莉老师为本课程的教学实践提供了帮助;中国科技大学微电子学教研室的赵天鹏、石林初、易波、谢家纯等教授为本书的撰写提供了指导和帮助;谢正茂、蔡志强、郭必成等同学和作者一道在北京多思集成电路中心从事过VLSI电路设计验证工作;李明、张松、丁文祥、吴雷、张文杰、董小波等同学在毕业论文中也从事了有关的研究工作并上机验证了本书部分程序。在此向以上人员表示感谢。在本书编写和教学实践中还参考、学习了有关数字系统设计、集成电路CAD方面多本书籍,在此向有关编著者致以深深的谢意。向一贯支持和关心作者的西安电子科技大学出版社的领导和杨宗周老师以及工作人员表示感谢。感谢一贯关心和支持我的读者朋友们。

有关本书的技术指导和建议请 E-mail to: jinx@ustc.edu.cn。

金 西
2003年1月于中国科技大学

目 录

第一章 复杂数字系统设计与 EDA 工具发展趋势	1
1.1 数字系统设计策略	1
1.1.1 数字系统自动设计的要求	1
1.1.2 VLSI 设计的描述域和层次	3
1.1.3 设计自动化	5
1.1.4 基于 VHDL 的 EDA 设计工具	6
1.2 数字系统的设计流程	8
1.2.1 Top-Down 设计流程	8
1.2.2 Bottom-Up 设计流程	9
1.2.3 正向设计和逆向设计	9
1.3 EDA 工具及其发展方向	11
1.3.1 选用适于复杂数字系统设计的 EDA 工具	11
1.3.2 与复杂数字系统有关 CAD 技术的发展趋势和难题	12
复习思考题	14
第二章 硬件描述语言 VHDL	15
2.1 VHDL 语言简介	15
2.1.1 VHDL 的特点	15
2.1.2 Verilog HDL 和 VHDL 的比较	16
2.1.3 传统设计与 VHDL 设计对照	18
2.1.4 VHDL 的新发展	20
2.2 VHDL 程序的基本结构	21
2.2.1 VHDL 程序的基本单元与构成	21
2.2.2 包(Package)、配置(Configuration)和库(Library)	27
2.2.3 设计实例	31
2.3 VHDL 语言的基本数据类型和操作符	33
2.3.1 数的类型和数的字面值	33
2.3.2 对象和分类	35
2.3.3 数据类型	36
2.3.4 运算操作符	39
2.4 VHDL 结构体的描述方式	39
2.4.1 顺序描述语句	40
2.4.2 并发描述语句	43
2.5 Active-VHDL 上机准备	46
2.5.1 Active-VHDL 的安装与启动	47

2.5.2	EDIT Plus 安装使用	47
2.5.3	熟悉 Active-VHDL 的集成环境	48
2.5.4	Active-VHDL 自带范例的调试流程	49
2.5.5	测试基准中的 VHDL 激励信号	49
2.5.6	Active-VHDL 中测试基准自动生成流程	50
2.5.7	半加器的波形分析	52
2.6	使用 VHDL 书写测试基准的方法	53
2.6.1	测试基准的作用	53
2.6.2	写激励所用的关键 VHDL 语句说明	54
2.6.3	激励的编写示例	54
2.7	基本逻辑电路的 VHDL 实现	58
2.7.1	组合逻辑电路设计	58
2.7.2	时序逻辑电路设计	62
2.8	VHDL 上机实践	70
2.8.1	VHDL 数字电路的文本描述、编译与仿真上机实验	70
2.8.2	交通灯控制器	71
	复习思考题	80

第三章 设计输入与仿真综合

3.1	复杂数字系统设计概述	81
3.1.1	复杂数字系统设计流程	81
3.1.2	EDA 硬件和软件平台配置	83
3.2	设计输入及其 EDA 工具软件	84
3.2.1	设计输入方式	84
3.2.2	功能仿真工具软件	85
3.2.3	VHDL 与 Verilog 相互转换	89
3.3	设计输入的形式	92
3.3.1	文本方式设计输入	92
3.3.2	Cadence 的设计输入流程	95
3.3.3	Cadence 电路图输入流程	96
3.4	VHDL 仿真建模	97
3.4.1	仿真的目的和方法	97
3.4.2	按 VHDL 的性能和功能建模	99
3.4.3	逻辑仿真模型	101
3.4.4	仿真系统及发展	105
3.5	设计综合	107
3.5.1	综合层次	108
3.5.2	逻辑综合	109
3.5.3	高层次综合	110
3.5.4	版图综合	112
3.5.5	综合的发展趋势	114
	复习思考题	115

第四章 EDA 技术与可编程 ASIC 的设计实现	116
4.1 集成电路与 ASIC	116
4.1.1 集成电路及其分类	116
4.1.2 ASIC 及其分类	117
4.2 可编程 ASIC 及其发展	118
4.2.1 可编程逻辑器件的发展历程	119
4.2.2 FPGA 和 CPLD 的比较	119
4.2.3 PLD/FPGA 入门	120
4.2.4 可编程 ASIC 的发展趋势	121
4.3 MAX+PLUS II 及其应用	124
4.3.1 MAX+PLUS II 软/硬件环境	125
4.3.2 MAX+PLUS II 的设计过程	127
4.3.3 用 MAX+PLUS II 设计交通灯控制器	128
4.3.4 基于 MAX+PLUS II 综合的实验	143
4.4 Xilinx 公司器件与 Foundation 应用	149
4.4.1 Xilinx 的 FPGA/CPLD 简介	149
4.4.2 FPGA/PLD 结构与原理	150
4.4.3 Xilinx 主流芯片简介	154
4.4.4 Xilinx XC9500 器件选用	156
4.4.5 Xilinx Foundation 开发系统设计流程简介	158
4.5 Xilinx XC95 系列在电压信号测量 A/D 中的应用	158
4.5.1 XC95108 简介	158
4.5.2 Xilinx Foundation 下的设计流程	160
4.6 CPLD/FPGA 在数字系统设计中的应用	178
4.6.1 Xilinx 网上设计方案的学习	178
4.6.2 Gold 码简介	179
4.6.3 DS-SS 系统中的 PN 序列	179
4.6.4 LFSR 的实现结构	180
4.6.5 Gold 码发生器	181
4.6.6 VHDL 源程序	182
复习思考题	187
第五章 版图编辑与版图验证	188
5.1 版图设计基础	188
5.1.1 版图设计方法简介	188
5.1.2 版图设计规则	190
5.2 Tanner Research Tools 组成与功能	194
5.2.1 安装并熟悉 L-edit pro 9.0/8.xx 版	194
5.2.2 安装 DOS 版 L-edit 5.0	196
5.2.3 版图编辑实践	196
5.2.4 读 CMOSLIB.TDB 的方法	197
5.2.5 L-edit 模块介绍	198

5.2.6	L-edit 主菜单使用导引	198
5.2.7	DRC 文件实例	199
5.3	版图设计流程和方法研究	201
5.3.1	IC Craftsman 版图软件介绍	201
5.3.2	布局设计	202
5.3.3	布线设计	203
5.3.4	制作器件版图	204
5.3.5	形成最小成分版图	205
5.3.6	版图集成	207
5.3.7	版图检测与调整	207
5.4	版图生成、验证	207
5.4.1	DataPath 设计	207
5.4.2	版图输入流程	213
5.4.3	MUX2 的版图编辑步骤	215
5.4.4	Diva 流程	220
5.4.5	Dracula 流程	221
5.4.6	参数提取反标	223
5.4.7	门级时序分析	224
5.4.8	晶体管级时序分析	228
	复习思考题	233

第六章 建库设计

6.1	建库及库信息	234
6.1.1	建库技术的新发展	234
6.1.2	使用 SUMMIT 工具的建库流程	238
6.1.3	库的总体信息	238
6.2	CMOS 电路基本器件的设计	243
6.2.1	CMOS 反相器的开关特性	243
6.2.2	CMOS 反相器设计准则	246
6.2.3	某工艺 CMOS 电路计算公式	246
6.2.4	CMOS 基本器件实例	248
6.3	电路仿真	260
6.3.1	Cadence 电路图 Spectre/Hspice 仿真流程	260
6.3.2	Hspice 仿真流程及激励编写规范	261
6.3.3	库单元的 LVS 检查	270
	复习思考题	273

第七章 Pspice 与器件模型

7.1	Pspice 实例分析	274
7.1.1	Spice 简介	274
7.1.2	Pspice 简介	275
7.1.3	Pspice 的使用	275
7.1.4	Pspice 模拟实例	279

7.2 器件模型应用	281
7.2.1 Pspice 中的器件模型	281
7.2.2 基于 Pspice 的 C 代码建模	287
7.2.3 Dracula LPE 寄生电容提取操作流程	292
复习思考题	294
第八章 测试技术与物理仿真	295
8.1 测试向量的生成	295
8.1.1 测试的基本概念	295
8.1.2 故障仿真	297
8.1.3 测试生成的过程	298
8.1.4 测试流程	300
8.2 可测试性设计	300
8.2.1 可测试性设计初步	301
8.2.2 可测试性设计与结构测试	304
8.2.3 软/硬件系统可测试性设计	307
8.2.4 包含嵌入式模块的可测试性设计	310
8.3 物理仿真	312
8.3.1 物理仿真的方法	312
8.3.2 芯片的 FPGA 物理仿真方法	313
8.3.3 混合的物理仿真	314
8.4 测试技术的新发展	317
8.4.1 可测性设计和测试技术的发展趋势	318
8.4.2 SoC 测试的发展趋势	319
8.4.3 测试方法学	320
复习思考题	321
附录一 VHDL 上机作业模板范例	322
附录二 参考网址	323
参考文献	324

第一章 复杂数字系统设计与 EDA 工具发展趋势

集成电路从 20 世纪 60 年代开始,经历了小规模(SSI)、中规模(MSI)、大规模(LSI)到目前的超大规模(VLSI)和特大规模(ULSI)阶段。1958 年设计出来的第一块集成电路只有 4 个晶体管,而目前已能制作规模达亿个晶体管以上的单个芯片,并构成一个完整的数字系统或数模混合的电子系统。几乎在集成电路的每个设计环节和整个设计过程都普遍使用 EDA(电子设计自动化)技术。要在几十平方毫米的硅片上完成线宽只有零点几个微米的上百万个器件的整个电子系统设计,只靠手工设计是完全不可能的,必须借用设计自动化(DA)或者 CAD 技术。

本章将介绍数字系统设计策略、设计流程以及 EDA 工具的发展趋势和选用指南。

1.1 数字系统设计策略

任何复杂的数字系统都可以最终分解成基本门和存储器元件,这种分解最好由计算机自动进行。VLSI 设计过程就是把高级的系统描述最终转化成如何生产芯片的描述过程,为了完成这样的转换,人们研究出了描述集成电子系统的抽象方法,这就是层次化、结构化的方法。层次化的设计方法,能使复杂的电子系统简化,并能在不同的设计层次及时发现错误并加以纠正;结构化的设计方法,能把复杂抽象的系统划分成一些可操作的模块,允许多个设计者同时设计,而且某些子模块的资源可以共用。

复杂的数字系统,不论是用 VLSI 或现场可编程门阵列(FPGA, Field Programming Gate Array)实现,还是用印刷电路板(PCB, Printed Circuit Board)实现,较好的策略都是用层次设计与自动设计相结合的方法。基本的设计过程是采用自顶向下(Top - Down)的设计,也就是说,从一个行为概念开始,建立越来越详细的层次结构,直至得到一个充分低的级,它能直接变换成物理实现,最后,物理实现完成整个数字系统的功能。

1.1.1 数字系统自动设计的要求

设计方法选取的主要依据是设计周期、设计成本、芯片成本、芯片尺寸、设计灵活性、保密性和可靠性等。

1. 设计正确性

设计的正确性是数字系统设计中最基本的要求。设计一旦完成并送交制造厂生产后,再发现有错误,就需要重新制版,重新流片。一个复杂的数字系统设计,电路、版图数据量大,要作一次修改,代价是非常昂贵的。目前专用集成电路(ASIC, Application Specific

Integrated Circuit)一般要求一次投片成功。中小规模集成电路可以人工验证,而对于 VLSI 来说,花费大量人工也无法保证设计的正确性。这就要求:在一个完整设计自动化系统的支持下,在各设计层次上都要进行反复验证和检查,各层次的设计数据都能自动转换和统一处理。由于数字系统设计的限制,需要有功能更强,性能更好的 EDA 设计工具将整个集成电路设计过程统一考虑,前后呼应,使系统设计达到最优。

目前,计算机辅助设计软件及工具几乎渗透到 VLSI 设计的各个步骤中,比如工艺模拟、器件模拟、电路分析、逻辑验证、版图验证及参数提取、布图工具、综合工具、计算机辅助设计、封装工具……。

在复杂数字系统设计中,测试是一个十分重要的课题。测试的意义在于检查电路是否能按设计要求正常工作。随着芯片功能的日趋复杂,测试费用所占的比例明显增大,虽然芯片测试是在芯片生产过程当中进行的,但是为了减小测试所需要的资源,往往在电路设计阶段就要充分考虑其可测试性的问题。具体做法是在已有的逻辑设计基础上添加一些专门用于测试的辅助电路。

2. 设计周期

由于市场竞争的需要,IC 产品要求几周甚至几天就要设计出来。在以往的芯片设计中,版图设计花费的时间最多。1978 年出品的微处理器 Z8000 的设计就是一例,它含有 17 500 个晶体管,版图设计花费了约 6600 个人时,占整个设计时间的 50%以上。但今天,随着设计水平的提高,目前一个上百万门的常规数字系统的芯片设计,一个设计团队半个月就可完成设计、验证工作。

3. 设计成本

每个芯片的成本 C_T 可以由下式计算:

$$C_T = \frac{C_D}{V} + \frac{C_P}{yn}$$

式中, C_D 为开发费用, C_P 为每片硅片的工艺成本, y 为平均成品率, V 为生产数量, n 为每片硅片上芯片的数目。

C_T 表明,对于小批量的产品,应着重减小开发费用 C_D ;对大批量的产品,应增加成品率 y 和增加每一硅片上的芯片数 n 。提高每一硅片上的芯片数主要靠提高工艺水平,减小小芯片尺寸,增大硅晶片面积(目前主流为 12 英寸硅片)来实现。通常,小批量 ASIC 采用半定制电路或可编程器件技术,大批量 ASIC 则采用全定制电路技术。另外,要增加实际成品率,又必须减小每个芯片的尺寸,这就要求在高层次设计中优化电路结构,在版图设计中减少布局和布线中的所谓“死区”,提高芯片利用率。

4. 产品性能

IC 的性能主要取决于所选择的电路系统的体系结构、器件工艺结构和版图设计的质量。为了提高 IC 的速度,需要采用流水线体系结构,这样自动化便成了设计的主流。

综上所述,一个复杂数字系统的设计就是在保证产品质量的前提下,正确地选择 IC 体系结构、器件形式和工艺方案,同时要尽可能地减小芯片尺寸,降低设计成本和缩短设计

周期。

1.1.2 VLSI 设计的描述域和层次

电路的层次设计过程,可以用 D. D. Gajski 于 1983 年提出的“Y”图描述。层次化、结构化的“Y”描述方法如图 1-1 所示。在图 1-1 中三个互不相同的设计域由三条射线轴表示。这三个设计域是:行为域、结构域和物理域。每个域中有多个抽象的级,而且离中心越远则抽象程度愈高。

行为域从概念上描述一个特定的系统做些什么,要完成什么功能,通常它只表示系统的输入/输出间的函数关系;行为域的设计着眼于严密地规定逻辑部件。它根据逻辑部件的规格目标,以考虑给出什么样的输入信号序列、形成什么样的内部状态、发生什么样的输出信号序列等信息为中心进行设计。它对于用什么样的逻辑电路来实现其功能并不特别在意,只是去正确地定义逻辑部件所应完成的功能。行为域是复杂数字系统设计的出发点。

结构域从概念上描述实现某一功能的具体结构以及各组成部件是怎样连接在一起的,包括各个单元的详细的端口定义。结构域的设计常常以线路图(Schematic Diagram)或线网表(Netlist)的形式给出。线网表可以以元件为中心,也可以以线网(节点)为中心。结构域的设计表面上不反映电路的功能特征,其功能是通过各单元的功能及其相互驱动关系来体现的,它是一个芯片设计的具体电路表示。

物理域描述的是结构的物理实现,即怎样实际制造出一个满足一定的连接关系的结构并能实现所要求功能的具体几何实现,例如集成电路的版图、布线的几何描述、PCB 板的元件封装说明、布局布线等。物理域的设计常常与具体的电路工艺条件相关联。

每一个设计域都可以在不同的抽象层次上描述,图 1-1 中的同心圆表示不同的抽象层次,这些抽象层次从高到低通常包含的设计级别有:系统级、算法级、模块或功能块的寄存器传输级(RTL)、逻辑级、电路级。

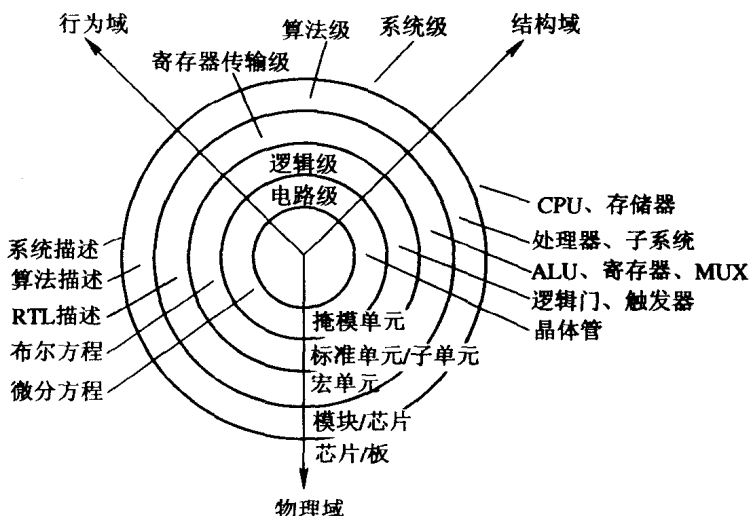


图 1-1 数字系统设计的“Y”图

通常，一个设计可以在三个设计域中进行描述，而根据设计的形式和电路的复杂性，可使用不同的抽象级别来表示。表 1-1 是对不同设计域和设计层次的总结。依靠 EDA 工具，由 RTL 级的行为域的描述自动转换成下一级(门级)的结构域描述，称为逻辑综合；由结构域的描述自动转换为物理域的描述，称为物理综合。这些综合技术是集成电路设计自动化中的关键技术。

表 1-1 VLSI 设计的层次描述

设计层次	行为域描述	结构域描述	设计考虑
系统级	自然语言描述的性能指标、结构	方框图	系统功能
算法级(芯片)	行为有限状态机、数据流图、控制流图	微处理器、存储器、串(并)行口、中断控制器	时序、同步、测试
寄存器级(宏单元)	数据流图、真值表、有限状态机、状态表、状态图	寄存器、ALU、计数器、MUX、ROM 等	时序、同步、测试
逻辑(门)级	布尔方程、卡诺图	逻辑门、器件(晶体管)	选用适当的基本门实现硬件
电路级	电压、电流的微分方程	晶体管、R、L、C 等	电路性能、延时、噪声
版图级	几何图形与工艺规则		

采用有效的设计方法是电路与系统设计成功的关键。设计过程的层次化、结构化把完整的硬件设计划分为一些可操作的模块，并允许多个设计者同时设计一个硬件系统中的不同模块，其中每个设计者负责自己所承担的部分。对于复杂数字系统设计，由于集成度太高，不能细分到电路级。因为 VLSI 芯片的结果非常强烈地依赖局部信息，而且整体性能涉及到局部设计，所以 VLSI 芯片设计是迭代的，要求频繁地调整，为此人们正在开发面向对象的 VHDL 来解决这种设计上的复杂性。

对于一个基于 $0.25\mu\text{m}$ 工艺流程的复杂数字系统设计，本书将给出设计的全部流程，包括前端设计、后端设计、建库设计和物理仿真等。对照“Y”图，一个数字系统的设计可简单描述如下：首先从用户用自然语言的描述开始构建要设计的数字系统的性能、结构规范等。用 VHDL 语言写行为级算法代码，通过综合程序将算法代码自动生成寄存器级 VHDL 代码、寄存器级逻辑电路图(这一步骤称高层次综合)，再通过综合程序自动生成(逻辑综合)门级的网表文件，至此，前端设计完成。当然，在前端设计中，工艺厂家的设计规则的导入及相应的仿真、建模和验证等步骤也是不可缺少的。其次，将生成的 Netlist 交由第三方 IP 整合服务公司处理，生成合乎某一流片工艺线需要的版图文件，经过版图生成、布局布线、版图后仿真等步骤，流片生成合格半定制芯片或全定制芯片。

1.1.3 设计自动化

复杂数字系统设计的复杂性,除了使设计周期延长外,还造成了设计人员的缺乏。从“Y”图可以看出,复杂数字系统的设计要求设计者不仅是一位电路设计者,而且是逻辑设计、计算机体系结构与应用软件的专家。为了解决芯片设计的危机,需要有新的电子设计自动化(EDA)方法,目前有 3 种方法在发展之中。

1. 计算机辅助的方法

计算机辅助的方法是最早的较为成熟的一种方法,也就是通常说的 CAD 技术。这种方法的基本思想是:所有的设计决定由设计者做出,设计危机可以通过提高设计者的工作效率来解决;给设计者提供完整的 EDA 工具软件,帮助克服设计的复杂性。这种方法是改良性的,因为它企图适应设计者目前的工作风格。设计者的传统工作习惯倾向于先设计出他自己的积木块(单元),然后用这些积木块作元件来实现更高级的结构,即所谓的自底向上(Bottom-Up)的方法。这种方法的优点是设计质量好,但费时间,且容易出错,需要多次迭代和比较。

2. 可编程的方法

可编程的方法就是利用编译的方法来设计数字系统。这种方法认为知识是算法的,而且可以写出变换程序,从问题的高级描述能自动生成或综合出它的全部或某些部分的解。可编程逻辑阵列(PLA)、复杂可编程逻辑器件(CPLD)以及 FPGA 都有很好的应用。本书第三章结合 VHDL 给出了具体完整的开发实例。这种自顶向下的方法是很好的方法,因为它替代了设计人员,而不只是在设计周期中帮助他们。可编程的方法在器件需求量大时,其性价比无法和全定制相比,但它可以作为全定制设计前的试制手段。

3. 智能的方法

智能的方法就是用人工智能方法设计复杂数字系统。这种方法认为,设计者的知识能存储到一个专家系统的知识库中。专家系统分为三类:一是概念,包括问题域中的基本术语,可从教科书中获得;二是规则,它描述特定的情景与希望完成的动作,这种知识是以经验为基础,是从专家那里得到的;三是策略,它是一些过程,辅助引导搜索知识库,而且当有几个同等合理的规则可使用时,可以帮助解决选择的矛盾。目前实用专家系统有专家布局器,专家布线器等。

目前,符合层次化、结构化设计的 EDA 系统的基本功能如图 1-2 所示。图 1-3 列举了 Cadence 和 Mentor Graphics 公司的通用数字系统设计软件包的主要模块。有关模块中术语的解释将在本书后续章节中介绍。

在通用数字系统设计中,用到的主要技术有 VHDL 语言、建模仿真、设计综合、可编程器件以及在深亚微米条件下的延迟计算、静态时序分析。

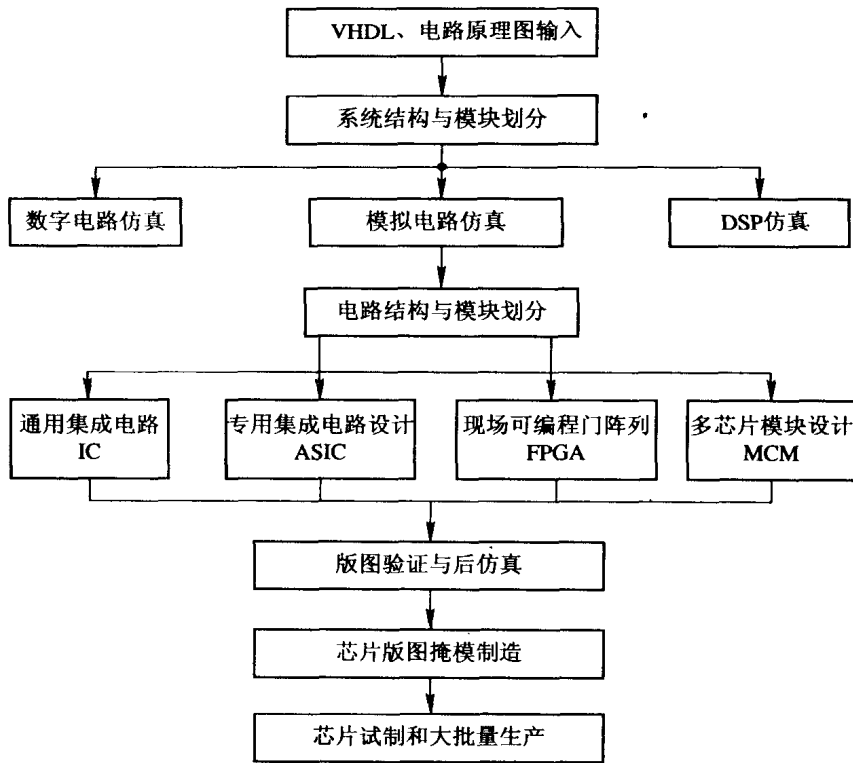


图 1-2 EDA 系统功能示意图

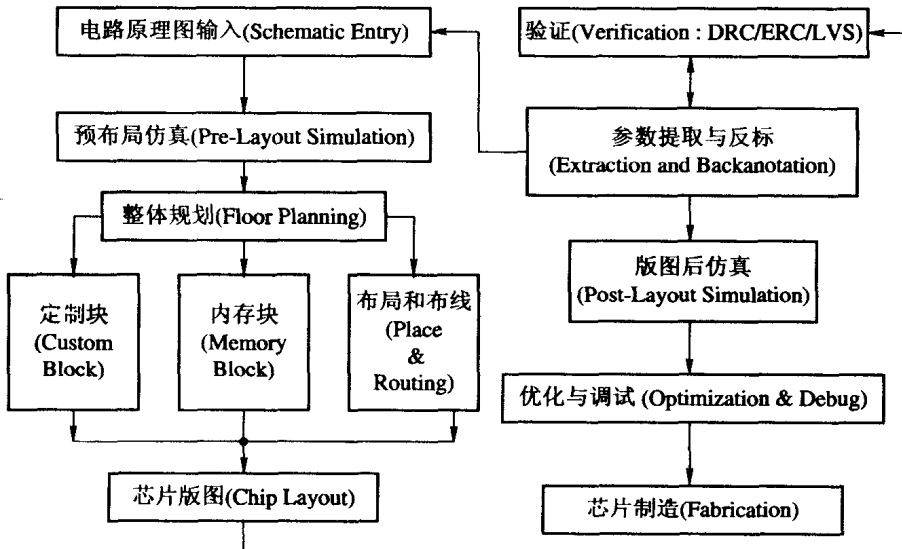


图 1-3 通用数字系统设计软件包的主要模块示意图

1.1.4 基于 VHDL 的 EDA 设计工具

随着半导体技术的迅速发展，在现代数字系统设计中，FPGA 和 CPLD 的使用越来越

广泛。与此同时,基于大规模可编程逻辑器件的 EDA 硬件解决方案也被广泛采用。一般来说,EDA 解决方案均采用计算机自顶向下的设计方式:在底层设计时对逻辑进行必要的描述,并依赖特定的软件执行逻辑优化(Logic Optimization)与器件映射(Device Mapping),最后再使用由各芯片生产厂商提供的编译器执行布线(Route)和网表优化(Netlist Optimization)。虽然对于简单的逻辑,采用原始逻辑图或布尔方程输入可以获得非常有效的结果,但对于复杂的数字系统设计,应用以上两种方案就很容易产生错误,而必须依靠一种高层次的逻辑输入方式。

为了适应自顶向下的设计方法,必须提供良好的自顶向下的设计环境。硬件描述语言(HDL, Hardware Description Language)提供了丰富的库和与工艺无关的设计输入方式,其中符合 IEEE-1076 标准的 VHDL 的应用成为新一代 EDA 解决方案中的首选。VHDL 的应用已成为当前以及未来 EDA 解决方案的核心,更是复杂数字系统的核心。

1. 应用 VHDL 的理由

严格地讲,VHDL 是一种用来描述数字逻辑系统的“编程语言”。它源于美国政府于 1980 年开始启动的超高速集成电路计划(VHSIC, Very High Speed Integrated Circuits)。在这一计划的执行过程中,专家们认识到需要有一种标准的语言来描述集成电路的结构和功能,由此,VHSIC 的硬件描述语言(VHSIC HDL),即 VHDL 便诞生了,并很快被美国电气和电子工程师协会(IEEE)所承认。

VHDL 是为了满足逻辑设计过程中的各种需求而设计的。首先,它能形式化地抽象表示电路的结构和行为,支持逻辑设计中层次与领域的描述。其二,支持电路描述由高层到低层的综合和转换,便于文档管理,易于理解和设计重用。最后,VHDL 具有电路仿真与验证功能,可以保证设计的正确性。用户甚至不必编写任何测试向量便可以进行源代码级的调试。而且,设计者可以非常方便地比较各种方案之间的可行性及其优劣,而不需做任何实际的电路实验。

VHDL 能够精确而且简明地描述数字电子系统,可用于从系统级到门级的描述,特别是能以非常抽象的形式反映出系统最本质的性能。VHDL 可为系统级模拟提供方便的手段,使得系统实现之前就可以评价系统的性能。鉴于 VHDL 具有以上诸多优点,只要开发者拥有 Pascal、C 等计算机高级语言的基础,便可以轻松地掌握 VHDL,使硬件工作软件化。

2. VHDL 的模块组织

在应用 VHDL 描述数字系统结构时,我们使用 Entity-Architecture(实体和结构)架构。

Entity 描述了数字系统的输入/输出接口,同时还定义了一些全局常量以及与其它电路(程序模块或逻辑图模块)之间必要连接的拓扑结构。但在 Entity 中,我们并不对电路的逻辑做任何描述,可将其看成是一个所谓的逻辑“黑盒子”。很明显,VHDL 遵循 EDA 解决方案中自顶向下的设计原则,并能够保持良好的接口兼容性。

Architecture 对具体的逻辑进行描述,主要由变量声明和语句描述两部分构成。变量有信号型和内部节点型;语句包括组合逻辑语句、同步时序逻辑语句。对于异步时序,则依靠语句排列顺序再加上延时语句来完成。

VHDL 在描述同步时序逻辑时主要使用 Process 结构和 Wait 语句。在一个结构体中可以同时拥有几个 Process，它是用于描述时序的基本单元。VHDL 描述硬件行为时，最重要的就是依据时间的变化对信号做出相应的分配和赋值，具体细节详见本书第二章。

在 EDA 解决方案中应用 VHDL 有助于缩短数字系统的开发周期。总的来讲，一般厂商均遵循 VHDL - 87 和 VHDL - 93 标准。

1.2 数字系统的设计流程

半导体技术和计算机技术的发展，使数字系统的设计理念和设计方法都发生了深刻的变化。以前，数字系统主要由一些固定功能的器件加上一定的外围电路构成模块，再由这些模块进一步形成各种功能电路。在设计时，几乎没有灵活性可言，而且，设计一个系统所需的芯片种类多且数目大。PLD 器件和 EDA 技术的广泛应用给今天的硬件设计者提供了强有力的支持，同时也使得数字系统的设计思想和设计方式发生了根本性的变化。一般来说，数字系统的设计有两种思路：一种是自顶向下(Top-Down)的设计思路，另一种是自底向上(Bottom-Up)的设计思路。

1.2.1 Top-Down 设计流程

随着集成电路技术的发展，电子系统的规模与复杂度越来越高，使用传统的自底向上的设计方法已越来越不适应，而自顶向下的设计方法却越来越显示出其优越性。

所谓自顶向下的设计方法如图 1-4 所示，它是指设计电子系统先从系统最抽象的层次出发，作高层次仿真，经过仿真验证后再经整体规划(Floor Planning)将系统行为操作分为子系统。各个子系统作行为仿真，它和高层次仿真的结果比较，易于发现和修正早期结

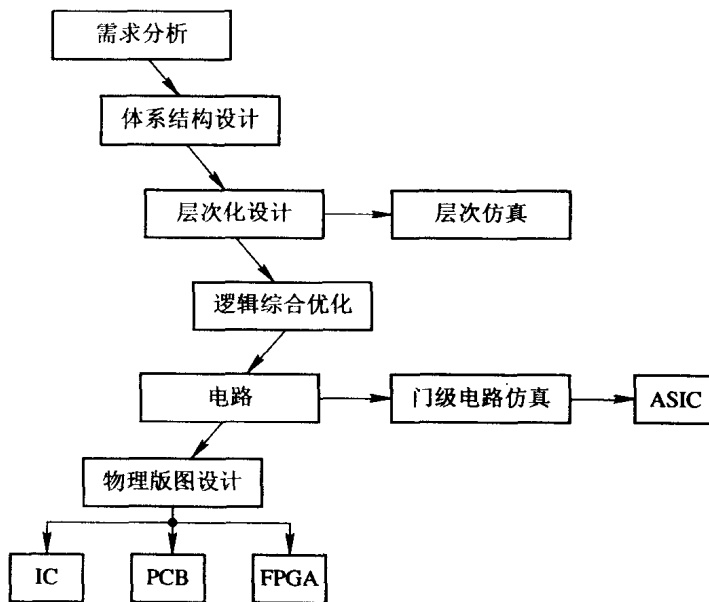


图 1-4 自顶向下(Top-Down)的设计方法