

高等学校电子信息类教材

数字系统设计 与Verilog HDL (第5版)

王金明 编著

<http://www.phei.com.cn>



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

· 014060290

TP332.1
05-5

高等学校电子信息类教材

数字系统设计与 Verilog HDL

(第5版)

王金明 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



北航

C1747490

TP332.1

05-5

内 容 简 介

本书根据 EDA 课程教学要求,以提高数字设计能力为目的,系统阐述 FPGA 数字系统开发的相关知识,主要内容包括 EDA 技术概述、FPGA/CPLD 器件、Verilog 硬件描述语言等。全书以 Quartus II、Synplify Pro 软件为平台,以 Verilog—1995 和 Verilog—2001 语言标准为依据,以可综合的设计为重点,通过大量经过验证的数字设计实例,阐述数字系统设计的方法与技术,由浅入深地介绍 Verilog 工程开发的知识与技能。

本书的特点是:着眼于实用,紧密联系教学实际,实例丰富。全书深入浅出,概念清晰,语言流畅。可作为电子、通信、微电子、信息、电路与系统、通信与信息系统及测控技术与仪器等专业本科生和研究生的教学用书,也可供从事电路设计和系统开发的工程技术人员阅读参考。

本书配有教学课件,可从华信教育资源网(www.hxedu.com.cn)免费下载。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

数字系统设计与 Verilog HDL / 王金明编著. —5 版. —北京: 电子工业出版社, 2014.7
高等学校电子信息类教材
ISBN 978-7-121-22537-6

I. ①数… II. ①王… III. ①数字系统—系统设计—高等学校—教材 ②硬件描述语言—程序设计—高等学校—教材 IV. ①TP271②TP312

中国版本图书馆 CIP 数据核字(2014)第 038219 号

责任编辑: 窦 昊 (douhao@phei.com.cn)

印 刷: 北京市李史山胶印厂

装 订: 北京市李史山胶印厂

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

开 本: 787×1 092 1/16 印张: 25.75 字数: 659 千字

版 次: 2002 年 1 月第 1 版

2014 年 7 月第 5 版

印 次: 2014 年 7 月第 1 次印刷

印 数: 4 000 册 定 价: 49.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

本书在第4版的基础上做了部分修订,增加了实验与设计的内容,使之更便于实践教学。本书的主要内容包括FPGA/CPLD器件、EDA设计工具(包括Quartus II、Synplify Pro、ModelSim等)、Verilog硬件描述语言,以及一些典型的数字设计实例。书中的实例多数在Altera的DE2或DE2—70实验板上做了实际验证,并尽量给出程序综合与仿真的结果,以便于对照。此外,对于EDA软件工具、实验平台、设计案例均做了精心选择,是作者认为目前较有典型性和代表性的方案。

本书的定位是作为EDA技术、FPGA开发或数字设计方面的教材。在编写的过程中,遵循的是重视基础、面向应用的原则,力图在有限的篇幅内,将EDA技术与FPGA设计相关的知识简明扼要、深入浅出地进行阐述,并融入作者在教学、科研中的实践经验。EDA技术是一门实践性的课程,所以选择一些具有趣味性的设计案例,有助于学生由浅入深快速地掌握设计语言,并通过实践加深体会。另外,本书与作者的另一拙作《数字系统设计与VHDL》(电子工业出版社,2010年5月)互为补充,前者以Verilog语言开发为主,后者则以VHDL语言的设计为重点。

全书共13章。第1章对EDA技术做了综述;第2章介绍FPGA/CPLD器件的结构与编程配置;第3章介绍Quartus II集成开发工具以及基于宏功能模块的设计;第4章对Verilog程序设计进行初步介绍;在第5、6章中,系统介绍Verilog的语法、语句、任务与函数等内容;第7章讨论Verilog设计的层次与风格;第8章是有关有限状态机的内容;第9章列举了一些典型电路的设计与实现;第10章讨论设计优化的问题;第11章是Verilog仿真与验证的内容;在第12章中就Verilog—2001、Verilog—2002标准对Verilog语言所做的扩展与增强做了较为全面的介绍;第13章是通信和信号处理等较为复杂的数字逻辑系统的设计举例。另外,在附录中除了给出Verilog HDL关键字和有关术语、缩略语之外,还对Altera的DE2和DE2—70实验平台做了介绍。

本书在撰写的过程中力求准确、简约,避免烦琐,以期做到深入浅出。所有举例均经过综合工具或仿真工具的验证。EDA技术是一门实践性很强的技术,同时EDA技术的发展又非常迅速,要真正掌握数字设计技术,成为行家里手,需要设计者在实践中不断摸索与积累,逐步提高自己的实际能力与水平。本书虽经很大努力,但由于编著者水平所限,书中疏漏与错误之处在所难免,希望同行和广大读者给予批评指正。

E-mail: wjm_ice@163.com

编著者

2014年5月于解放军理工大学

目 录

第 1 章 EDA 技术概述	1
1.1 EDA 技术及其发展	1
1.2 Top-down 设计与 IP 核复用	4
1.2.1 Top-down 设计	4
1.2.2 Bottom-up 设计	5
1.2.3 IP 复用技术与 SoC	5
1.3 数字设计的流程	7
1.3.1 设计输入	7
1.3.2 综合	9
1.3.3 布局布线	9
1.3.4 仿真	10
1.3.5 编程配置	10
1.4 常用的 EDA 软件工具	10
1.5 EDA 技术的发展趋势	14
习题 1	15
第 2 章 FPGA/CPLD 器件	16
2.1 PLD 器件概述	16
2.1.1 PLD 器件的发展历程	16
2.1.2 PLD 器件的分类	17
2.2 PLD 的基本原理与结构	19
2.2.1 PLD 器件的基本结构	19
2.2.2 PLD 电路的表示方法	20
2.3 低密度 PLD 的原理与结构	21
2.4 CPLD 的原理与结构	26
2.4.1 宏单元结构	26
2.4.2 典型 CPLD 的结构	27
2.5 FPGA 的原理与结构	30
2.5.1 查找表结构	30
2.5.2 典型 FPGA 的结构	32
2.6 FPGA/CPLD 的编程元件	37
2.7 边界扫描测试技术	41
2.8 FPGA/CPLD 的编程与配置	43

2.8.1	在系统可编程	43
2.8.2	CPLD 器件的编程	44
2.8.3	FPGA 器件的配置	44
2.9	FPGA/CPLD 器件概述	50
2.10	FPGA/CPLD 的发展趋势	54
	习题 2	55
第 3 章	Quartus II 集成开发工具	56
3.1	Quartus II 原理图设计	56
3.1.1	半加器原理图设计输入	56
3.1.2	编译与仿真	60
3.1.3	1 位全加器编译与仿真	64
3.2	Quartus II 的优化设置	65
3.2.1	分析与综合设置	65
3.2.2	优化布局布线	67
3.2.3	设计可靠性检查	72
3.3	Quartus II 的时序分析	73
3.3.1	时序设置与分析	73
3.3.2	时序逼近	75
3.4	基于宏功能模块的设计	77
3.4.1	Megafunctions 库	77
3.4.2	Maxplus2 库	82
3.4.3	Primitives 库	83
	习题 3	85
	实验与设计	87
3-1	简易数字钟	87
3-2	序列产生器	88
3-3	m 序列发生器	89
3-4	8 位带符号乘法器	89
3-5	模 24 方向可控计数器	92
3-6	用锁相环模块实现倍频和分频	94
第 4 章	Verilog 设计初步	97
4.1	Verilog 简介	97
4.2	Verilog 模块的结构	98
4.3	Verilog 基本组合电路设计	102
4.3.1	用 Verilog 设计基本组合电路	102
4.3.2	用 Verilog 设计加法器	102
4.4	Verilog 基本时序电路设计	105
4.4.1	用 Verilog 设计触发器	105
4.4.2	用 Verilog 设计计数器	106

习题 4	108
实验与设计	108
4-1 Synplify Pro 综合器的使用方法	108
4-2 Synplify 综合器的使用方法	112
第 5 章 Verilog 语法与要素	114
5.1 Verilog 语言要素	114
5.2 常量	115
5.2.1 整数 (Integer)	115
5.2.2 实数 (Real)	117
5.2.3 字符串 (Strings)	117
5.3 数据类型	118
5.3.1 net 型	118
5.3.2 variable 型	119
5.4 参数	121
5.5 向量	123
5.6 运算符	125
习题 5	129
实验与设计	130
5-1 RAM 存储器设计	130
5-2 用 rom 模块实现 4×4 无符号数乘法器	132
第 6 章 Verilog 行为语句	136
6.1 过程语句	136
6.1.1 always 过程语句	137
6.1.2 initial 过程语句	140
6.2 块语句	141
6.2.1 串行块 begin-end	141
6.2.2 并行块 fork-join	142
6.3 赋值语句	143
6.3.1 持续赋值与过程赋值	143
6.3.2 阻塞赋值与非阻塞赋值	144
6.4 条件语句	145
6.4.1 if-else 语句	146
6.4.2 case 语句	147
6.5 循环语句	151
6.5.1 for 语句	152
6.5.2 repeat、while、forever 语句	153
6.6 编译指示语句	155
6.7 任务与函数	156

6.7.1	任务 (task)	156
6.7.2	函数 (function)	158
6.8	顺序执行与并发执行	161
	习题 6	162
	实验与设计	163
6-1	4×4 矩阵键盘检测电路	163
6-2	计数器设计	164
第 7 章 Verilog 设计的层次与风格		167
7.1	Verilog 设计的层次	167
7.2	门级结构描述	167
7.2.1	Verilog HDL 内置门元件	168
7.2.2	门级结构描述	170
7.3	行为描述	171
7.4	数据流描述	172
7.5	不同描述风格的设计	173
7.5.1	半加器设计	173
7.5.2	1 位全加器设计	174
7.5.3	4 位加法器设计	176
7.6	多层次结构电路的设计	177
7.7	基本组合电路设计	179
7.7.1	门电路	179
7.7.2	编译码器	179
7.7.3	其他组合电路	181
7.8	基本时序电路设计	182
7.8.1	触发器	182
7.8.2	锁存器与寄存器	182
7.8.3	计数器与串并转换器	184
7.8.4	简易微处理器	185
7.9	三态逻辑设计	187
	习题 7	189
	实验与设计	189
7-1	数字表决器	189
7-2	FIFO 缓存器设计	192
第 8 章 Verilog 有限状态机设计		195
8.1	有限状态机	195
8.2	有限状态机的 Verilog 描述	197
8.2.1	用三个过程描述	198
8.2.2	用两个过程描述	199

8.2.3 单过程描述方式	201
8.3 状态编码	203
8.3.1 常用的编码方式	203
8.3.2 状态编码的定义	207
8.4 有限状态机设计要点	209
8.4.1 复位和起始状态的选择	209
8.4.2 多余状态的处理	209
习题 8	211
实验与设计	211
8-1 流水灯控制器	211
8-2 汽车尾灯控制器	213
8-3 状态机 A/D 采样控制电路	214
8-4 用状态机实现字符液晶显示控制	215
第 9 章 Verilog 设计进阶	222
9.1 加法器设计	222
9.1.1 级连加法器	222
9.1.2 数据流描述的加法器	223
9.1.3 超前进位加法器	224
9.1.4 流水线加法器	225
9.2 乘法器设计	226
9.2.1 并行乘法器	226
9.2.2 移位相加乘法器	228
9.2.3 布斯乘法器	231
9.2.4 查找表乘法器	233
9.3 奇数分频与小数分频	234
9.3.1 奇数分频	234
9.3.2 半整数分频与小数分频	235
9.4 VGA 图像的显示与控制	237
9.4.1 DE2—70 的 VGA 显示电路	237
9.4.2 VGA 图像显示原理与时序	238
9.4.3 VGA 图像显示与控制的实现	241
9.5 点阵式液晶显示控制	247
9.6 乐曲演奏电路	252
习题 9	257
实验与设计	259
9-1 数字跑表	259
9-2 实用多功能数字钟	266
第 10 章 Verilog 设计的优化	275
10.1 设计的可综合性	275
10.2 流水线设计技术	277

10.3	资源共享	281
10.4	过程	283
10.5	阻塞赋值与非阻塞赋值	285
	习题 10	289
	实验与设计	289
10-1	小数分频	289
10-2	如何在 FPGA 设计中消除毛刺	291
10-3	消抖动电路	294
第 11 章 Verilog 仿真与测试		295
11.1	系统任务与系统函数	295
11.2	用户自定义元件	299
11.2.1	组合电路 UDP 元件	300
11.2.2	时序逻辑 UDP 元件	301
11.3	延时模型的表示	303
11.3.1	时间标尺定义`timescale	303
11.3.2	延时的表示与延时说明块	304
11.4	测试平台	305
11.5	组合和时序电路的仿真	308
11.5.1	组合电路的仿真	308
11.5.2	时序电路的仿真	310
	习题 11	310
	实验与设计	311
11-1	用 ModelSim 仿真 8 位二进制加法器	311
11-2	仿真乘累加器	316
第 12 章 Verilog 语言的发展		318
12.1	Verilog—2001 语法结构	318
12.1.1	语法结构的扩展与增强	318
12.1.2	设计管理	324
12.1.3	系统任务和系统函数的扩展	326
12.1.4	VCD 文件的扩展	329
12.2	Verilog—2002 语法结构	330
12.2.1	硬件单元建模	331
12.2.2	属性	334
12.2.3	编程语言接口	338
	习题 12	339
第 13 章 通信与信号处理设计实例		340
13.1	m 序列发生器	340
13.1.1	m 序列的原理与性质	340
13.1.2	m 序列产生器设计	342

13.2	Gold 码	344
13.2.1	Gold 码的原理与性质	344
13.2.2	Gold 码产生器设计	345
13.3	CRC 校验码	347
13.4	FSK 解调	349
13.5	数字过零检测与等精度频率测量	352
13.5.1	数字过零检测法	352
13.5.2	等精度频率测量	353
13.6	QPSK 调制器的 FPGA 实现	356
13.7	FIR 数字滤波器	359
13.8	FPGA 信号处理基础及浮点计算实例	363
13.8.1	定点数的表示法	363
13.8.2	浮点数的表示法	364
13.8.3	定点数到浮点数的格式转换	366
13.8.4	浮点数乘法	368
13.8.5	浮点数加法	371
13.8.6	浮点数除法	374
	习题 13	376
	实验与设计	376
13-1	信号音产生器	376
13-2	异步串行接口 (UART)	383
附录 A	Verilog HDL (IEEE Std 1364—1995) 关键字	389
附录 B	Verilog HDL (IEEE Std 1364—2001) 关键字	390
附录 C	DE2 系统介绍	391
附录 D	DE2—70 系统介绍	393
附录 E	有关术语与缩略语	395
	参考文献	400

第 1 章 EDA 技术概述

我们已经进入数字化和信息化的时代，其特点是各种数字产品的广泛应用。现代数字产品在性能提高、复杂度增大的同时，更新换代的步伐也越来越快，实现这种进步的因素在于芯片制造技术和设计技术的进步。

芯片制造技术以微细加工技术为代表，目前已进展到深亚微米阶段，可以在几平方厘米的芯片上集成数千万个晶体管。摩尔曾经对半导体集成技术的发展做出预言：大约每 18 个月，芯片的集成度提高 1 倍，功耗下降 1 倍，他的预言被人们称为摩尔定律（Moore's law）。几十年来，集成电路的发展与这个预言非常吻合，数字器件经历了从 SSI, MSI, LSI 到 VLSI，直到现在的 SoC（System on Chip，芯片系统），我们已经能够把一个完整的电子系统集成在一个芯片上。还有一种器件的出现极大改变了设计制作电子系统的方式与方法，这就是可编程逻辑器件（Programmable Logic Device, PLD）。PLD 器件是 20 世纪 70 年代后期发展起来的一种器件，它经历了可编程逻辑阵列（Programmable Logic Array, PLA）、通用阵列逻辑（Generic Array Logic, GAL）等简单形式到现场可编程门阵列（Field Programmable Gate Array, FPGA）和复杂可编程逻辑器件（Complex Programmable Logic Device, CPLD）的高级形式的发展，它的广泛使用不仅简化了电路设计，降低了研制成本，提高了系统可靠性，而且给数字系统的整个设计和实现过程带来了革命性的变化。

电子系统的设计理念和设计方法也发生了深刻的变化，从电子 CAD（Computer Aided Design）、电子 CAE（Computer Aided Engineering）到电子设计自动化（Electronic Design Automation, EDA），设计的自动化程度越来越高，设计的复杂性也越来越强。

EDA 技术已成为现代电子设计技术的有力工具，没有 EDA 技术的支持，要完成超大规模集成电路的设计和制造是不可想象的，反过来，生产制造技术的进步又不断对 EDA 技术提出新的要求，促使其不断向前发展。

1.1 EDA 技术及其发展

在现代数字系统的设计中，EDA 技术已经成为一种普遍的工具。对设计者而言，熟练地掌握 EDA 技术，可以极大地提高工作效率，起到事半功倍的效果。

EDA（电子设计自动化）技术没有一个精确的定义，我们可以这样来认识，所谓的 EDA 技术就是以计算机为工作平台，以 EDA 软件工具为开发环境，以 PLD 器件或者 ASIC 专用集成电路为目标器件设计实现电路系统的一种技术。EDA 技术的发展以计算机科学、微电子技术的发展为基础，并融合了应用电子技术、智能技术，以及计算机图形学、拓扑学、计算数学等众多学科的最新成果发展而来的。EDA 技术经历了一个由简单到复杂、由初级到高级不断发展进步的阶段。从 20 世纪 70 年代，人们就已经开始基于计算机开发出一些软件工具帮助设计者完成电路系统的设计任务，以代替传统的手工设计方法，随着计算机软件和硬件技术水平的提高，EDA 技术也不断进步，大致经历了下面三个发展阶段。

1. CAD 阶段

电子 CAD 阶段是 EDA 技术发展的早期阶段(时间大致为 20 世纪 70 年代至 80 年代初)。在这个阶段,一方面,计算机的功能还比较有限,个人计算机还没有普及;另一方面,电子设计软件的功能也较弱。人们主要是借助计算机对所设计电路的性能进行一些模拟和预测;另外,就是完成 PCB 的布局布线,简单版图的绘制等工作。

2. CAE 阶段

集成电路规模的扩大,电子系统设计的逐步复杂,使得电子 CAD 的工具逐步完善和发展,尤其是人们在设计方法学、设计工具集成化方面取得了长足的进步,EDA 技术就进入了电子 CAE 阶段(时间大致为 20 世纪 80 年代初至 90 年代初)。在这个阶段,各种单点设计工具、各种设计单元库逐渐完备,并且开始将许多单点工具集成在一起使用,大大提高了工作效率。

3. EDA 阶段

20 世纪 90 年代以来,微电子工艺有了显著的发展,工艺水平已经达到了深亚微米级,在一个芯片上已经可以集成数目上千万乃至上亿的晶体管,芯片的工作速度达到了 Gbps 级,这样就对电子设计的工具提出了更高的要求,也促进了设计工具的发展。

在今天,EDA 技术已经成为电子设计的普遍工具,无论是设计集成电路还是设计普通的电子电路,没有 EDA 工具的支持,都是难以完成的。EDA 技术的使用包括电子工程师进行电子系统开发的全过程,以及进行开发设计涉及的各个方面。从一个角度看,EDA 技术可粗略分为系统级、寄存器传输级(RTL)、门级和版图级几个层次的辅助设计过程;从另一个角度来看,EDA 技术包括电子电路设计的各个领域,即从低频电路到高频电路、从线性电路到非线性电路、从模拟电路到数字电路、从 PCB 设计到 FPGA 开发等,EDA 技术的功能和范畴如图 1.1 所示。

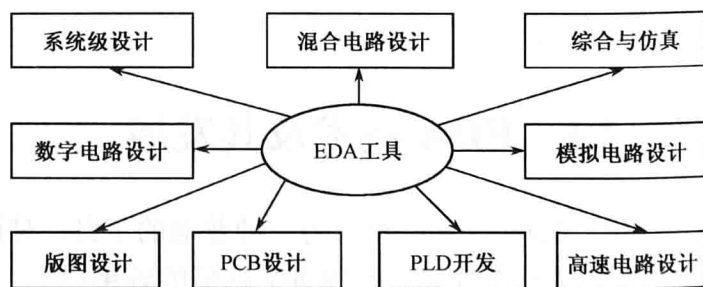


图 1.1 EDA 技术的功能和范畴

进入 21 世纪后,EDA 技术得到更快的发展,开始步入一个新的时期,突出地表现在以下几个方面。

(1) 电子技术各个领域全方位融入 EDA 技术,除日益成熟的数字技术外,可编程模拟器件的设计技术也有了很大的进步。EDA 技术使得电子领域各学科的界限更加模糊,相互包容和渗透,如模拟与数字、软件与硬件、系统与器件、ASIC 与 FPGA、行为与结构等,软硬件协同设计技术也成为 EDA 技术的一个发展方向。

(2) IP (Intellectual Property) 核在电子设计领域得到广泛的应用,基于 IP 核的 SoC

(System on Chip, 芯片系统) 设计技术趋向成熟, 电子设计成果的可重用性进一步提高。

(3) 嵌入式微处理器软核的出现, 更大规模的 FPGA/CPLD 器件的不断推出, 使得 SoPC (System on Programmable Chip, 可编程芯片系统) 步入实用化阶段, 在一片 FPGA 芯片中实现一个完备的系统成为可能。

(4) 用 FPGA (Field Programmable Gate Array, 现场可编程门阵列) 器件实现完全硬件的 DSP (数字信号处理) 处理成为可能, 用纯数字逻辑进行 DSP 模块的设计, 使得高速 DSP 实现成为现实, 并有力地推动了软件无线电技术的实用化。基于 FPGA 的 DSP 技术为高速数字信号处理算法提供了实现途径。

(5) 在设计和仿真两方面支持标准硬件描述语言的 EDA 软件不断推出, 系统级、行为验证级硬件描述语言的出现 (如 System C) 使得复杂电子系统的设计和验证更加高效。在一些大型的系统设计中, 设计验证工作非常艰巨, 这些高效的 EDA 工具的出现, 减轻了开发人员的工作量。

除了上述的发展趋势, 现代 EDA 技术和 EDA 工具还呈现出以下一些共同的特点。

1) 采用硬件描述语言 (HDL) 进行设计

采用硬件描述语言 (Hardware Description Language, HDL) 进行电路与系统的描述是当前 EDA 设计技术的另一个特征。与传统的原理图设计方法相比, HDL 语言更适合于描述规模大、功能复杂的数字系统, 它能够使设计者在比较抽象的层次上对所设计系统的结构和逻辑功能进行描述。采用 HDL 语言进行设计的突出优点是: 语言的标准化, 便于设计的复用、交流、保存和修改; 设计与工艺的无关性, 宽范围的描述能力, 便于组织大规模、模块化的设计。目前最常用的硬件描述语言是 Verilog HDL 和 VHDL, 它们都已成为 IEEE 标准。

2) 逻辑综合与优化

目前的 EDA 工具最高只能接受行为级 (Behavior Level) 或寄存器传输级 (Register Transport Level, RTL) 描述的 HDL 文件进行逻辑综合, 并进行逻辑优化。为了能更好地支持自顶向下的设计方法, EDA 工具需要在更高的层级进行综合和优化, 这样可进一步缩短设计周期, 提高设计效率。

3) 开放性和标准化

现代 EDA 工具普遍采用标准化和开放性的框架结构, 可以接纳其他厂商的 EDA 工具一起进行设计工作。这样可实现各种 EDA 工具间的优化组合, 并集成在一个易于管理的统一环境之下, 实现资源共享, 有效提高设计者的工作效率, 有利于大规模、有组织的设计开发工作。

4) 更完备的库 (Library)

EDA 工具要具有更强大的设计能力和更高的设计效率, 必须配有丰富的库, 比如元器件图形符号库、元器件模型库、工艺参数库、标准单元库、可复用的电路模块库、IP 库等。在电路设计的各个阶段, EDA 系统需要不同层次、不同种类的元器件模型库的支持。例如, 原理图输入时需要原理图符号库、宏模块库, 逻辑仿真时需要逻辑单元的功能模型库, 模拟电路仿真时需要模拟器件的模型库, 版图生成时需要适应不同层次和不同工艺的底层版图库等。各种模型库的规模和功能是衡量 EDA 工具优劣的一个重要标志。

总而言之, 从过去发展的过程看, EDA 技术一直滞后于制造工艺的发展, 它在制造技术

的驱动下不断进步;从长远看,EDA 技术将随着微电子技术、计算机技术的不断发展而发展。“工欲善其事,必先利其器”,EDA 工具在现代电子系统的设计中所起的作用越来越大,未来它将在诸多因素的推动下继续进步。

1.2 Top-down 设计与 IP 核复用

数字系统的设计方法发生了深刻的变化。传统的数字系统采用搭积木式的方式进行设计,即由一些固定功能的器件加上一定的外围电路构成模块,由这些模块进一步形成各种功能电路,进而构成系统。构成系统的积木块是各种标准芯片,如 74/54 系列(TTL)、4000/4500 系列(CMOS)芯片等,这些芯片的功能是固定的,用户只能根据需要从这些标准器件中选择,并按照推荐的电路搭成系统。在设计时,几乎没有灵活性可言,设计一个系统所需的芯片种类多且数量大。

PLD 器件和 EDA 技术的出现,改变了这种传统的设计思路,使人们可以立足于 PLD 芯片来实现各种不同的功能,新的设计方法能够由设计者自己定义器件的内部逻辑和引脚,将原来由电路板设计完成的工作大部分放在芯片的设计中进行。这样不仅可以通过芯片设计实现各种数字逻辑功能,而且由于引脚定义的灵活性,减轻了原理图和印制板设计的工作量和难度,增加了设计的自由度,提高了效率。同时这种设计减少了所需芯片的种类和数量,缩小了体积,降低了功耗,提高了系统的可靠性。

在基于 EDA 技术的设计中,通常有两种设计思路:一种是自顶向下的设计思路,另一种是自底向上的设计思路。

1.2.1 Top-down 设计

Top-down 设计,即自顶向下的设计。这种设计方法首先从系统设计入手,在顶层进行功能方框图的划分和结构设计。在功能级进行仿真、纠错,并用硬件描述语言对高层次的系统行为进行描述,然后用综合工具将设计转化为具体门电路网表,其对应的物理实现可以是 PLD 器件或专用集成电路(ASIC)。由于设计的主要仿真和调试过程是在高层次上完成的,这一方面有利于早期发现结构设计上的错误,避免设计工作的浪费,同时也减少了逻辑功能仿真的工作量,提高了设计的一次成功率。

在 Top-down 的设计中,将设计分成几个不同的层次:系统级、功能级、门级和开关级等,按照自上而下的顺序,在不同的层次上,对系统进行设计与仿真。图 1.2 是这种设计方式的示意图。如图中所示,在 Top-down 的设计过程中,需要有 EDA 工具的支持,有些步骤 EDA 工具可以自动完成,比如综合等,有些步骤 EDA 工具为用户提供了操作平台。Top-down 的设计必须经过“设计—验证—修改设计—再验证”的过程,不断反复,直到得到的结果能够完全实现所要求的逻辑功能,并且在速度、功耗、价格和可靠性方面实现较为合理的平衡。不过,这种设计也并非绝对的,在设计的过程中,有时也需要用到自下而上的方法,就是在系统划分和分解的基础上,先进行底层单元设计,然后再逐步向上进行功能块、子系统的设计,直至构成整个的系统。

如图 1.3 所示是用 Top-down 的设计方式设计 CPU 的示意图。首先在系统级划分,将整个 CPU 划分为几个模块,如 ALU, PC, RAM 模块等,对每个模块再分别进行设计与描述,然后通过 EDA 工具将整个设计综合为门级网表,并实现它。在设计过程中,需要进行多次仿真和验证,不断修改设计。

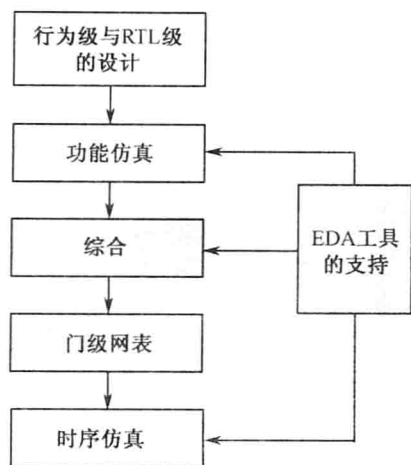


图 1.2 Top-down 设计方式

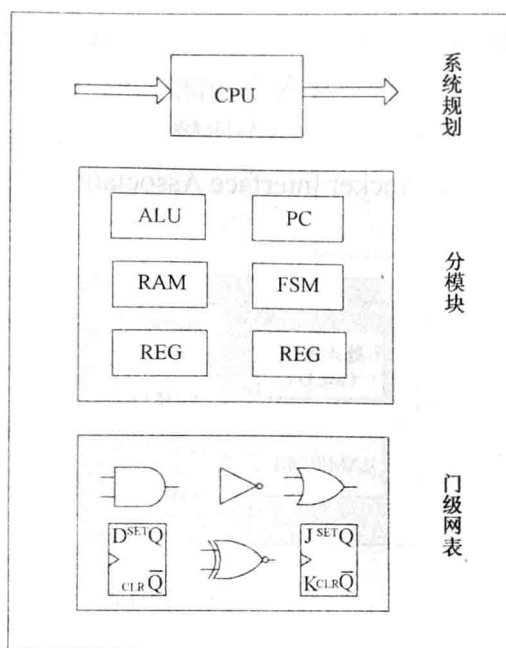


图 1.3 CPU 的 Top-down 设计方式示意图

1.2.2 Bottom-up 设计

Bottom-up 设计，即自底向上的设计，这是一种传统的设计思路。这种设计方式，一般是设计者选择标准集成电路，或者将各种基本单元，如各种门电路以及加法器、计数器等模块做成基本单元库，调用这些基本单元，逐级向上组合，直到设计出满足自己需要的系统为止。这样的设计方法就如同一砖一瓦建造金字塔，不仅效率低、成本高，而且容易出错。

Top-down 的设计由于更符合人们逻辑思维的习惯，也容易使设计者对复杂的系统进行合理的划分与不断的优化，因此是目前设计思想的主流。而 Bottom-up 的设计往往使设计者关注了细节，而对整个的系统缺乏规划，当设计出现问题时，如果要修改，就会比较麻烦，甚至前功尽弃，不得不从头再来。因此，在数字系统的设计中，主要采用 Top-down 的设计思路，而以 Bottom-up 设计为辅。

1.2.3 IP 复用技术与 SoC

当电子系统的设计越来越向高层发展时，基于 IP 复用 (IP Reuse) 的设计技术越来越显示出优越性。IP (Intellectual Property)，其原来的含义是指知识产权、著作权等，在 IC 设计领域可将其理解为实现某种功能的设计，IP 核 (IP 模块) 则是指完成某种功能的设计模块。

IP 核分为硬核、固核和软核三种类型。软核指的是在寄存器级或门级对电路功能用 HDL 进行描述，表现为 VHDL 或 Verilog HDL 代码，软核与生产工艺无关，不涉及物理实现，为后续设计留有很大的空间，增大了 IP 的灵活性和适应性。用户可以对软核的功能加以裁剪以符合特定的应用，也可以对软核的参数进行设置，包括总线宽度、存储器容量、使能或禁止功能块等。硬核指的是以版图形式实现的设计模块，它基于一定的设计工艺，通常用 GDS II 格式表示，不同的客户可以根据自己的需要选用特定生产工艺下的硬核。固核是完成了综合的功能块，通常以网表的形式提交客户使用。软核使用灵活，但其可预测性差，延时不一定能达到要求；硬核可靠性高，能确保性能，如速度、功耗等，能够很快地投入使用。

如图 1.4 所示，由微处理器核 (MPU Core)，数字信号处理器核 (DSP Core)，存储器核

(RAM/ROM), A/D、D/A 核以及 USB 接口核等构成一个单片系统 (SoC)。用户在设计一个系统时,可以自行设计各个功能模块,也可以用 IP 模块来构建。作为设计者来说,想要在短时间内开发出新产品,一个比较好的方法就是使用 IP 核完成设计。目前,还有专门的组织 VSIA (Virtual Socket Interface Association, 虚拟插座接口协会) 来制定关于 IP 产品的标准与规范。

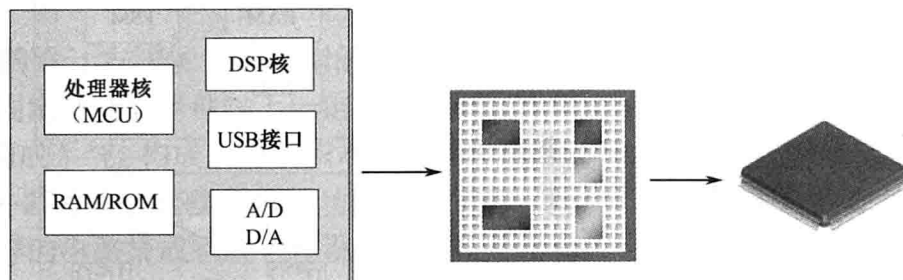


图 1.4 系统芯片 (SoC) 示意图

如上所述,基于 IP 复用的开发给设计者带来了诸多好处,如节省时间、缩短开发周期、避免重复劳动等。当然,IP 的发展还存在一些问题,比如 IP 版权的保护、IP 的保密以及 IP 间的集成等。但基于 IP 复用的设计技术无疑会成为电子系统开发的重要手段之一。

系统芯片 (SoC),或者称为芯片系统、片上系统,是指把一个完整的系统集成在一个芯片上;或者说是用一个芯片实现一个功能完整的系统。系统芯片可以采用全定制的方式来实现,把设计的网表文件提交给半导体厂家流片就可得到,但采用这种方式风险性高,费用大,周期长。还有一种方式就是采用可编程逻辑器件来实现。CPLD 和 FPGA 的集成度越来越高,速度也越来越快,设计者可以在其上通过编程完成自己的设计。今天,已不仅能用它们实现一般的逻辑功能,还可以把微处理器、DSP、存储器、标准接口等功能部件全部集成在其中,真正实现 System on Chip。

微电子制造工艺的进步为 SoC 的实现提供了硬件基础,而 EDA 软件技术的提高则为 SoC 创造了必要的开发平台。目前,EDA 的新工具、新标准和新方法正在向着高层化发展,过去已将设计从晶体管级提高到了逻辑门级,后来,又提高到了寄存器传输级,现在则越来越多地在系统级完成。

在数字系统进入 SoC 时代后,设计方法也随之而产生变化。如果把器件的设计看成是设计者根据设计规则用软件来搭接已有的不同模块,那么早期的设计是基于晶体管的 (Transistor Based Design)。在这一阶段,设计者最关心的是怎样减小芯片的面积,所以又称为面积驱动的设计 (Area Driving Design, ADD)。随着设计方法的改进,出现了以门级模块为基础的设计 (Gate Based Design)。在这一阶段,设计者在考虑芯片面积的同时,更多关注门级模块之间的延时,所以这种设计又称为时延驱动的设计 (Time Driving Design, TDD)。20 世纪 90 年代以来,芯片的集成度进一步提高,系统芯片 SoC 的出现,使得以 IP 模块复用为基础的设计逐渐流行,这种设计方法称为基于模块的设计方法 (Block Based Design, BBD)。在应用 BBD 方法进行设计的过程中,逐渐产生的一个问题是,在开发完一个产品后,怎么能尽快开发出其系列产品。这样就产生了新的概念——PBD, PBD 是基于平台的设计方法 (Platform Based Design),它是一种基于 IP 的,面向特定应用领域的 SoC 设计环境,可以在更短的时间内设计出满足需要的电路。PBD 的实现依赖于如下关键技术的突破:如高层次