

EDA工具应用丛书

数字系统设计 与Verilog HDL (第3版)

王金明 编著

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



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

EDA 工具应用丛书

数字系统设计与 Verilog HDL

(第3版)

王金明 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统介绍了数字系统设计相关的知识, 主要内容包括: EDA 技术、FPGA/CPLD 器件、Verilog 硬件描述语言等。本书以 Quartus II、Synplify Pro/Synplify 软件为平台, 以 Verilog-1995 和 Verilog-2001 为语言标准, 以可综合的设计为重点, 以大量经过验证的数字设计实例为依据, 系统阐述了数字系统设计的方法与技术, 对设计优化做了探讨。

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、信息、测控、电路与系统等专业高年级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。

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. —3 版. —北京: 电子工业出版社, 2009.1

(EDA 工具应用丛书)

ISBN 978-7-121-07990-0

I. 数… II. 王… III. ①数字系统—系统设计 ②硬件描述语言, VHDL—程序设计 IV. TP271 TP312

中国版本图书馆 CIP 数据核字(2009)第 199119 号

责任编辑: 雷洪勤

印 刷: 北京牛山世兴印刷厂
装 订:

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: 787×1092 1/16 印张: 21.75 字数: 557 千字

印 次: 2009 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 39.80 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 zllts@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

本书第1版和第2版承蒙广大读者的认可和喜爱，并被许多院校选做课程教材，作者在前两版的基础上，对全书的内容做了全面修订和改写。第3版较之第2版，在以下几个方面做了比较大的改进：

(1) 开发工具选择以 Quartus II、Synplify Pro/Synplify 软件为主，去掉了 MAX+plus II 软件的有关内容；

(2) 设计语言在 Verilog-1995 的基础上，对 Verilog-2001、Verilog-2002 两个标准也做了系统的介绍；

(3) 增加了基于宏功能模块设计的内容；

(4) 增加了更多典型的数字设计开发实例；

(5) FPGA/CPLD 器件更注重基本原理与知识的系统阐述。

本书的所有实例均在实验平台上进行了验证，并尽量给出程序综合与仿真的结果，以便于对照，并有助于建立语言描述和硬件电路的映射关系。读者也可将这些设计移植到其他软件平台。

目前，EDA 技术已成为电子信息类学生一门重要的专业基础课程，并在教学、科研，以及大学生电子设计竞赛等活动中，起着越来越重要的作用，成为电子信息类本科生及研究生必须掌握的基础知识与基本技能。随着 EDA 技术的应用日益广泛，对 EDA 课程教学的要求也不断提高，必须对教学内容进行优化和更新，以与 EDA 技术的发展相适应。正是基于以上考虑，修订了本书的内容。

全书共 12 章，主要包括以下内容。

第 1 章对 EDA 技术作了综述；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；第 4 章是基于宏功能模块的设计。在第 5、6、7 章中，系统介绍了 Verilog 的语法、语句、任务与函数等内容。第 8、9 章列举了常用数字电路的设计与实现；第 10 章讨论了设计优化的问题；第 11 章是有关电路仿真的内容；第 12 章是算法和较复杂数字逻辑系统的设计举例。在附录 A、B 中就 Verilog-2001、2002 标准对 Verilog 语言所做的扩展和增强做了较为全面的介绍。

王金明编写了本书大部分章节，并对全书进行了统稿，冷自强编写了第 4 章，研究生黄建军、王耿、闵刚、潘新星等对部分程序进行了调试，张雄伟教授对本书进行了审阅，并提出了修改意见，在此一并表示感谢。

本书在修订的过程中，遵循的是重视基础、面向应用这样的原则。EDA 技术是不断向前发展的，要掌握数字设计技术，需设计者在实践中不断摸索与积累，逐步提高自己的设计水平，本书虽经很大努力，由于时间急促及作者水平所限，书中错误与疏漏之处在所难免，希望同行和广大读者给予批评指正。

作者的电子邮件地址：wjm_ice@163.com

本书可免费提供电子教学课件。

作者

2008 年 10 月

第 2 版前言

计算机技术和微电子工艺的发展,使得现代数字系统的设计和应用进入了新的阶段。电子设计自动化(EDA)技术在数字系统设计中起的作用越来越重要,新的工具和新的设计方法不断推出,可编程逻辑器件不断增加新的模块,功能越来越强,硬件设计语言也顺应形势,推出新的标准,更加好用,更加便捷。

本书第 1 版承蒙广大读者的认可和喜爱,并被许多学校选为教材,推出三年多时间以来,由于技术的发展,需要对书中的内容进行更新,因此对全书认真做了全面的修订。第 2 版较之前一版,在以下几个方面做了比较大的改进:

(1) 由于 FPGA 器件的更新换代非常快,因此对有关器件的内容根据其发展做了更新。

(2) 在软件设计方面,由于 Quartus II 软件的使用越来越广泛,因此增加了 Quartus II 的使用介绍;而 MAX+plus II 发展到 10.2 版本后,Altera 不再推出新版本,但考虑到 MAX+plus II 由于其方便易用和优良的性能,仍受到很多设计人员的喜爱和欢迎,在很多学校也仍在使用,是经典的大众化设计工具,因此仍保留 MAX+plus II,并增加了若干典型实例,同时增加了 Synplify Pro/Synplify、ModelSim 等第三方工具软件使用方法的介绍。

(3) 由于原理图设计开发也是很重要的数字设计手段,同时在科研开发中被广泛采用,因此增加了原理图设计一章。

(4) 在 Verilog HDL 语言方面做了全面的扩充,对 Verilog HDL 数字系统设计做了更为深入的阐述,以可综合的设计为主,增加了 Verilog HDL 仿真的内容。

(5) 增加了实验设计部分,通过 10 个典型的实验让读者熟悉原理图和文本设计输入方式,熟悉软件的使用和有关的设计技术,便于实验教学。

(6) 增加了习题等内容。

由于 EDA 技术的发展,使得现代数字系统设计从设计思想、设计工具到实现方式都产生了深刻的变化,呈现出新的特点,怎样在教学中体现出这些变化,抓住 EDA 技术和数字设计的实质,是每一个在该学科领域从事教学的老师所面临和思考的问题。其根本目的是在有限的学时内,将数字设计技术本质的内容作为讲授的重点,传达给学生,从而掌握数字系统设计和 EDA 技术的精髓。

本书主要以 CPLD/FPGA 器件、EDA 软件工具、Verilog HDL 硬件描述语言三方面内容作为主线,上述三者是进行现代数字系统设计不可或缺的工具,也是每个设计者必须掌握的内容。

首先数字系统的设计和实现离不开 CPLD/FPGA 器件,因为很多数字系统是基于 CPLD/FPGA 器件实现的,本书对 CPLD/FPGA 器件的结构与功能特点做了分析,对目前主流的 PLD 器件做了概述。

EDA 软件工具也在设计开发中起着重要作用,好的 EDA 软件为数字设计开发提供了平台和工具,它将设计者的设计思想自动、高效地转化为物理电路或网表结构,并以直观、便捷的形式提供了仿真模拟的手段。没有 EDA 工具的支持,进行现代数字设计是不可想像的,本书详细介绍了典型 EDA 工具的使用方法,并以典型实例的设计作为展示。

此外,在现代数字系统设计中,一个明显的特点是越来越多地采用硬件描述语言来进行设计,作为一个从事数字设计开发的工程师,必须至少熟练掌握一种 HDL 语言,其中

Verilog HDL 和 VHDL 显然是首选,这两种语言已成为 IEEE 的标准,并各有自己的特色。本书选择 Verilog HDL 语言作为讲授的重点,Verilog HDL 语言是在 C 语言基础上发展而来的硬件描述语言,具有简捷、高效、易学、功能强的特点,具有广泛的应用群体。本书系统介绍 Verilog HDL 语言的语法、语句,以可综合的设计为重点,通过大量的例程由浅入深地展示了 Verilog HDL 数字设计的方法与技巧。所有实例全部通过了调试,并尽量给出例程综合和仿真的结果,以便于对照,并有助于建立语言描述和硬件电路的映射关系,也可以将这些设计移植到其他设计平台上。

本书共 12 章。第 1 章对 EDA 技术进行综述,解释了有关的概念,介绍流行的 EDA 软件和数字系统设计的流程。第 2 章介绍典型 CPLD/FPGA 器件的结构与特点,以及在系统编程(ISP)和边界扫描测试(BST)技术,并对目前主流的 CPLD/FPGA 器件做了概述。第 3 章中以具体的实例,介绍用 Quartus II、MAX+plus II 软件进行原理图设计开发的过程。第 4 章介绍基于硬件描述语言的设计过程以及与文本设计有关的典型 EDA 软件工具的使用方法。在第 5、6 章中,详细介绍了 Verilog HDL 的语言要素、行为语句、任务和函数等内容。第 7 章介绍 Verilog HDL 数字设计的层次和描述风格。第 8 章是有关仿真的内容,包括系统任务和函数、UDP 元件、延时模型等,还提供电路仿真的具体实例。第 9 章列举了常用数字电路的设计方法。第 10 章讨论了设计方法和设计优化的问题。第 11 章是算法和较复杂数字逻辑的实现方法。第 12 章通过 10 个典型的实验进一步熟悉原理图和文本设计输入方式,熟悉典型软件的使用方法。

感谢张雄伟教授对作者的支持与帮助,感谢杨吉斌博士的大力协助。

数字系统设计与 EDA 技术是不断发展变化的,要掌握数字设计和 EDA 技术的精髓,需要设计者在不断实践的过程中不懈地摸索和积累,逐步提高自己的设计水平。本书试图起到快速入门、抛砖引玉的作用,但由于作者水平所限,虽然做了很大努力,书中错误与疏漏之处仍在所难免,真诚希望同行和广大读者给予批评指正。

作者的电子邮件地址: wjm_ice@163.com。

目 录

第 1 章 EDA 技术与数字系统设计	(1)
1.1 EDA 技术及其发展	(1)
1.2 数字系统设计技术	(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 数字系统设计的流程	(7)
1.3.1 输入	(8)
1.3.2 综合	(10)
1.3.3 适配	(10)
1.3.4 仿真	(11)
1.3.5 编程	(11)
1.4 常用的 EDA 软件工具	(11)
1.5 EDA 技术的发展趋势	(15)
习题	(16)
第 2 章 FPGA/CPLD 器件	(17)
2.1 PLD 的分类	(17)
2.1.1 按集成度分类	(18)
2.1.2 按编程特点分类	(19)
2.1.3 按结构特点分类	(19)
2.2 PLD 的基本原理与结构	(20)
2.2.1 PLD 器件的基本结构	(20)
2.2.2 PLD 电路的表示方法	(21)
2.3 低密度 PLD 的原理与结构	(22)
2.4 CPLD 的原理与结构	(27)
2.4.1 宏单元结构	(27)
2.4.2 典型 CPLD 的结构	(28)
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)
2.5.3 FPGA 结构的发展	(36)
2.6 FPGA/CPLD 的编程元件	(37)
2.6.1 熔丝型开关	(37)
2.6.2 反熔丝	(38)

2.6.3 浮栅编程元件	(38)
2.6.4 SRAM 编程元件	(41)
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 在系统编程	(43)
2.8.2 CPLD 器件的编程	(44)
2.8.3 FPGA 器件的配置	(45)
2.9 FPGA/CPLD 器件概述	(50)
2.10 PLD 的发展趋势	(53)
习题	(54)
第 3 章 Quartus II 集成开发工具	(55)
3.1 Quartus II 原理图设计	(55)
3.1.1 半加器原理图输入	(55)
3.1.2 半加器设计与仿真	(59)
3.1.3 全加器设计与仿真	(63)
3.2 Quartus II 的优化设置	(64)
3.2.1 Settings 设置	(64)
3.2.2 分析与综合设置	(66)
3.2.3 优化布局布线	(66)
3.2.4 设计可靠性检查	(72)
3.3 Quartus II 的时序分析	(75)
3.3.1 时序设置与分析	(75)
3.3.2 时序逼近	(77)
习题	(80)
第 4 章 基于宏功能模块的设计	(85)
4.1 乘法器模块	(85)
4.2 除法器模块	(88)
4.3 计数器模块	(90)
4.4 常数模块	(92)
4.5 锁相环模块	(94)
4.6 存储器模块	(97)
4.7 其他模块	(101)
习题	(103)
第 5 章 Verilog HDL 设计初步	(105)
5.1 Verilog HDL 简介	(105)
5.2 Verilog HDL 设计举例	(106)
5.3 Verilog HDL 模块的结构	(109)
5.4 Synplify pro/Synplify 综合器	(112)

习题	(119)
第 6 章 Verilog HDL 语法与要素	(120)
6.1 Verilog HDL 语言要素	(120)
6.2 常量	(121)
6.3 数据类型	(124)
6.3.1 net 型	(124)
6.3.2 variable 型	(125)
6.4 参数	(127)
6.5 向量	(127)
6.6 运算符	(129)
习题	(134)
第 7 章 Verilog HDL 行为语句	(135)
7.1 过程语句	(135)
7.1.1 always 过程语句	(136)
7.1.2 initial 过程语句	(139)
7.2 块语句	(140)
7.2.1 串行块 begin-end	(140)
7.2.2 并行块 fork-join	(141)
7.3 赋值语句	(142)
7.3.1 持续赋值与过程赋值	(142)
7.3.2 阻塞赋值与非阻塞赋值	(143)
7.4 条件语句	(144)
7.4.1 if-else 语句	(144)
7.4.2 case 语句	(145)
7.5 循环语句	(148)
7.5.1 for 语句	(148)
7.5.2 repeat、while、for 语句	(149)
7.6 编译指示语句	(151)
7.7 任务与函数	(153)
7.7.1 任务 (task)	(153)
7.7.2 函数 (function)	(155)
7.8 顺序执行与并发执行	(158)
习题	(159)
第 8 章 数字设计的层次与风格	(162)
8.1 数字设计的层次	(162)
8.2 结构描述	(162)
8.2.1 Verilog HDL 内置门元件	(163)
8.2.2 门级结构描述	(165)

8.3	行为描述	(166)
8.4	数据流描述	(167)
8.5	不同描述风格的设计	(168)
8.6	基本组合电路设计	(172)
8.6.1	门电路	(172)
8.6.2	编译码器	(174)
8.6.3	数据选择器	(176)
8.6.4	其他组合电路	(176)
8.7	基本时序电路设计	(177)
8.7.1	触发器	(177)
8.7.2	锁存器与寄存器	(178)
8.7.3	计数器与串并转换器	(180)
8.7.4	简易微处理器	(181)
	习题	(182)
第 9 章	Verilog HDL 设计进阶	(184)
9.1	加法器设计	(184)
9.1.1	级连加法器	(184)
9.1.2	并行加法器	(185)
9.1.3	超前进位加法器	(186)
9.1.4	流水线加法器	(188)
9.2	乘法器设计	(188)
9.2.1	并行乘法器	(189)
9.2.2	移位相加乘法器	(191)
9.2.3	加法树乘法器	(193)
9.2.4	查找表乘法器	(195)
9.3	乘累加器 (MAC)	(195)
9.4	奇数分频与小数分频	(197)
9.4.1	奇数分频	(197)
9.4.2	半整数分频与小数分频	(198)
9.5	数字跑表	(201)
9.6	数字频率计	(202)
9.7	交通灯控制器	(205)
9.8	乐曲演奏电路	(208)
9.9	实用多功能数字钟	(212)
	习题	(216)
第 10 章	数字设计的优化	(218)
10.1	设计的可综合性	(218)
10.2	流水线设计技术	(220)
10.3	资源共享	(223)

10.4	有限状态机 (FSM) 设计	(226)
10.4.1	基于状态机的设计	(226)
10.4.2	基于状态机的设计要点	(231)
10.5	多层次结构电路的设计	(233)
10.6	进程	(236)
10.7	阻塞赋值与非阻塞赋值	(237)
10.8	FPGA 设计中毛刺的消除	(241)
	习题	(244)
第 11 章	数字电路的仿真	(246)
11.1	系统任务与系统函数	(246)
11.2	用户自定义元件 (UDP)	(250)
11.3	延时模型的表示	(254)
11.4	数字电路的仿真	(256)
11.4.1	测试平台 (Test Bench)	(256)
11.4.2	组合电路的仿真	(259)
11.4.3	时序电路的仿真	(262)
	习题	(263)
第 12 章	数字设计实例	(264)
12.1	基于 FPGA 实现点阵式液晶显示控制	(264)
12.1.1	GDM12864 的结构及指令	(264)
12.1.2	显示数据提取与设计源代码	(265)
12.2	基于 CPLD 实现字符液晶显示控制	(269)
12.2.1	字符液晶 HS162 结构及指令	(269)
12.2.2	状态机设计及源代码	(272)
12.3	VGA 彩条信号发生器	(275)
12.3.1	VGA 显示原理	(276)
12.3.2	源码及仿真	(277)
12.4	VGA 图像显示控制器设计	(279)
12.5	异步串行接口 (UART) 设计	(281)
12.5.1	UART 工作原理	(282)
12.5.2	源码及仿真	(283)
12.6	直接数字频率合成器 (DDS)	(285)
12.6.1	DDS 原理与顶层设计	(285)
12.6.2	分模块设计	(287)
12.6.3	使用嵌入式逻辑分析仪进行在线测试	(291)
12.7	等精度频率计	(294)
12.8	FIR 数字滤波器	(296)
12.9	CRC 校验码	(299)
	习题	(302)

附录 A	Verilog-2001 语法结构.....	(304)
附录 B	Verilog-2002 语法结构.....	(317)
附录 C	Verilog HDL (IEEE Std 1364-1995) 关键字	(325)
附录 D	Verilog HDL (IEEE Std 1364-2001) 关键字	(326)
附录 E	Quartus II 支持的 Verilog HDL 结构	(327)
附录 F	Synplify Pro/Synplify 可综合的 Verilog 结构	(329)
附录 G	有关术语与缩略语.....	(330)

第 1 章 EDA 技术与数字系统设计

我们已经进入了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，其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作出预言：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，系统芯片），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）技术没有一个很精确的定义，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。人们主要是借助于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

随着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,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上千万乃至上亿的晶体管,芯片的工作速度达到了 Gbps,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。

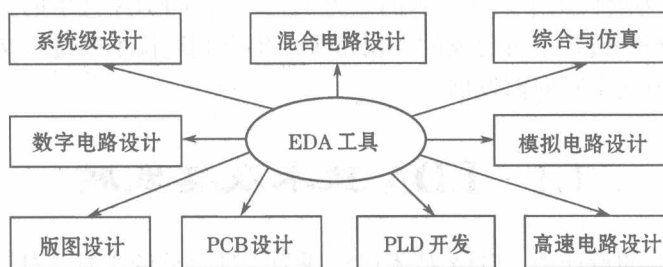


图 1.1 EDA 技术的功能和范畴

进入到 21 世纪后,EDA 技术得到了更快的发展,开始步入了一个新的时期,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,除了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了广泛的应用,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, 有效提高了设计者的工作效率, 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 技术一直是滞后于制造工艺的发展的, 它在制造技术的驱动下, 不断地向前进步; 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 数字系统设计技术

数字系统的设计方法发生了深刻的变化。传统的数字系统通常是采用搭积木式的方式设计的,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,而且由于管脚定义的灵活性,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,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、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。

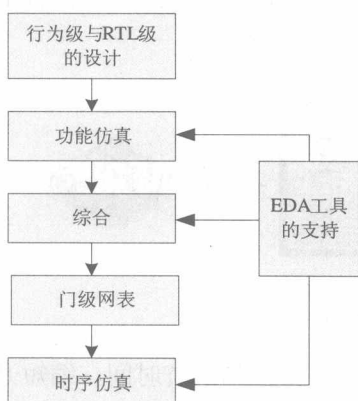


图 1.2 Top-down 设计方式

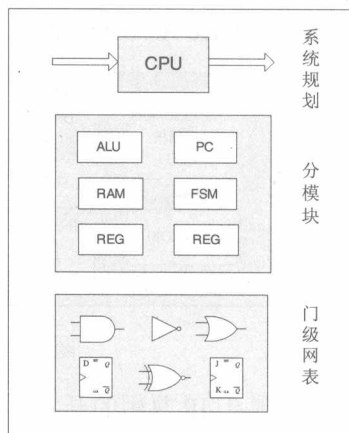


图 1.3 CPU 的 Top-down 设计方式示意图

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，如果要修改的话，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的时候，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