



EDA 工具应用丛书

数字系统设计 与 Verilog HDL (第2版)

王金明 编著



EDA 工具应用丛书

数字系统设计与 Verilog HDL

(第2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

本书按照“器件—软件—设计语言”的顺序介绍数字系统设计的方法、CPLD/FPGA 器件、典型的 EDA 设计软件和 Verilog HDL 硬件描述语言，力求涵盖数字系统开发涉及的主要技术，并在内容上进行取舍，做了精心的编排。

本书以具体的实例，详细介绍用 Quartus II、MAX+plus II 软件进行原理图设计和文本设计开发的过程，对 Synplify Pro/Synplify、ModelSim 等专用开发工具的使用与相互接口也做了介绍。

Verilog HDL 数字设计是本书的重点，以可综合的设计为重点，同时对仿真和模拟也做了深入阐述，讨论了设计方法和设计优化的问题，以大量经过验证的数字设计实例为依据，全面系统地阐述了 Verilog HDL 硬件设计开发的方法与技巧。

本书突出的特点是：着眼于实用性，紧密联系教学和科研实际，实例丰富。全书概念清晰，语言流畅，可读性强。书中加入了大量的图和表，以增强表述的效果。

本书可作为电子工程、通信工程、仪器仪表、数字信号处理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。

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. —2 版. —北京: 电子工业出版社, 2005.9

(EDA 工具应用丛书)

ISBN 7-121-01494-7

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. ①TP271 ②TP312

中国版本图书馆 CIP 数据核字 (2004) 第 071879 号

责任编辑: 雷洪勤

印 刷: 北京天宇星印刷厂

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

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 24.25 字数: 618 千字

印 次: 2005 年 9 月第 1 次印刷

印 数: 5 000 册 定价: 35.00 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 zits@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

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)
内容提要	(1)
1.1 引言	(1)
1.2 EDA 技术及其发展	(2)
1.3 数字系统的设计技术	(3)
1.3.1 Top-down 设计	(4)
1.3.2 Bottom-up 设计	(5)
1.3.3 IP 复用技术与 SoC	(5)
1.4 数字系统的实现方式	(6)
1.5 数字系统的设计输入方式	(7)
1.6 数字系统设计的流程	(9)
1.6.1 设计输入	(9)
1.6.2 综合	(10)
1.6.3 适配	(11)
1.6.4 仿真	(12)
1.6.5 编程	(12)
1.7 常用的 EDA 设计工具	(12)
1.7.1 集成的 CPLD/FPGA 开发工具	(12)
1.7.2 设计输入工具	(13)
1.7.3 逻辑综合器	(13)
1.7.4 仿真器	(14)
1.7.5 版图设计软件及其他常用 EDA 工具	(15)
习题	(16)
第 2 章 CPLD/FPGA 的结构与配置	(17)
内容提要	(17)
2.1 概述	(17)
2.2 PLD 的分类	(18)
2.2.1 PLD 的集成度分类	(18)
2.2.2 按编程特点分类	(19)
2.2.3 按结构特点分类	(20)
2.3 PLD 的基本结构	(20)
2.3.1 PLD 器件的基本结构	(20)
2.3.2 PLD 电路的表示方法	(21)
2.3.3 简单 PLD 的结构	(22)

2.4	CPLD 的结构与特点	(24)
2.4.1	Lattice 的 CPLD 结构特点	(25)
2.4.2	Altera 的 CPLD 结构特点	(28)
2.5	FPGA 的结构与特点	(33)
2.5.1	Xilinx 的 FPGA 结构特点	(33)
2.5.2	Altera 的 FPGA 结构特点	(36)
2.6	在系统编程与边界扫描测试技术	(43)
2.6.1	在系统编程技术 (ISP)	(44)
2.6.2	边界扫描测试 (BST) 技术	(45)
2.7	CPLD/FPGA 器件产品概述	(47)
2.7.1	Lattice 的 CPLD/FPGA 器件	(47)
2.7.2	Xilinx 的 CPLD/FPGA 器件	(49)
2.7.3	Altera 的 CPLD/FPGA 器件	(53)
2.8	CPLD/FPGA 器件的配置	(58)
2.8.1	ByteBlaster 及其使用	(58)
2.8.2	CPLD 器件的配置	(59)
2.8.3	FPGA 器件的配置	(60)
2.9	PLD 器件的发展趋势	(65)
	习题	(65)
第 3 章	原理图设计输入方式	(66)
	内容提要	(66)
3.1	原理图设计的流程	(66)
3.2	Quartus II 原理图设计	(69)
3.2.1	半加器原理图输入	(69)
3.2.2	半加器编译	(73)
3.2.3	半加器仿真	(75)
3.2.4	全加器设计与仿真	(78)
3.3	MAX+plus II 原理图设计	(79)
3.3.1	原理图设计输入	(80)
3.3.2	编译与手动调整	(84)
3.3.3	仿真与时间特性分析	(88)
3.3.4	编程下载	(91)
3.4	基于 LPM 宏单元库的设计	(94)
3.4.1	LPM 宏单元库	(94)
3.4.2	LPM 设计举例	(96)
	习题	(97)
第 4 章	基于硬件描述语言的设计	(100)
	内容提要	(100)
4.1	基于 HDL 文本输入的设计流程	(100)

4.2	Quartus II 文本设计举例	(101)
4.2.1	创建工程文件	(101)
4.2.2	编译	(103)
4.2.3	仿真	(105)
4.3	MAX+plus II 文本设计举例	(106)
4.4	Synplify pro 文本设计举例	(109)
4.4.1	用 Synplify Pro 综合的过程	(110)
4.4.2	Synplify Pro 与 MAX+plus II 的接口	(114)
4.4.3	Synplify Pro 与 Quartus II 的接口	(116)
4.5	Synplify 文本设计举例	(116)
4.6	ModelSim 仿真举例	(119)
4.6.1	ModelSim 功能仿真	(120)
4.6.2	ModelSim 时序仿真	(125)
4.7	ispLEVER 文本设计举例	(126)
	习题	(129)
第 5 章	Verilog HDL 数字设计初步	(130)
	内容提要	(130)
5.1	Verilog HDL 语言的历史及与 C 语言的比较	(130)
5.2	Verilog HDL 设计举例	(132)
5.2.1	4 位全加器和 4 位计数器	(132)
5.2.2	综合与仿真	(133)
5.3	Verilog HDL 模块的基本结构	(137)
5.4	Verilog HDL 语言要素	(140)
5.4.1	空白符和注释	(140)
5.4.2	标志符、运算符和关键字	(141)
5.5	常量	(141)
5.6	数据类型和变量	(144)
5.6.1	连线型	(145)
5.6.2	寄存器型	(146)
5.6.3	parameter	(147)
5.7	寄存器和存储器	(147)
5.8	运算符	(148)
5.8.1	运算符	(149)
5.8.2	运算符的优先级	(152)
	习题	(153)
第 6 章	Verilog HDL 行为语句	(154)
	内容提要	(154)
6.1	概述	(154)
6.2	过程语句	(155)

6.2.1	always 过程语句	(155)
6.2.2	initial 语句	(159)
6.3	块语句	(160)
6.3.1	串行块 begin-end	(160)
6.3.2	并行块 fork-join	(161)
6.4	赋值语句	(162)
6.4.1	持续赋值与过程赋值	(162)
6.4.2	阻塞赋值与非阻塞赋值	(163)
6.5	条件语句	(164)
6.5.1	if-else 语句	(164)
6.5.2	case 语句	(165)
6.5.3	条件语句使用要点	(168)
6.6	循环语句	(169)
6.6.1	for 语句	(169)
6.6.2	repeat 语句	(170)
6.6.3	while 和 forever 语句	(171)
6.7	编译向导	(173)
6.7.1	宏替换'define	(173)
6.7.2	文件包含#include	(174)
6.7.3	条件编译'ifdef、'else、'endif	(175)
6.8	任务和函数	(175)
6.8.1	任务 (task)	(175)
6.8.2	函数 (function)	(178)
6.8.3	任务和函数的区别	(181)
6.9	顺序执行与并发执行	(182)
6.10	Verilog-2001 语法结构简介	(184)
	习题	(187)
第 7 章	Verilog HDL 数字设计的层次与风格	(188)
	内容提要	(188)
7.1	Verilog HDL 数字设计的层次	(188)
7.2	结构描述	(189)
7.2.1	Verilog HDL 内置门元件	(189)
7.2.2	门级结构描述	(191)
7.3	行为描述	(192)
7.4	数据流描述	(193)
7.5	不同描述风格的设计	(195)
7.5.1	半加器	(195)
7.5.2	1 位全加器	(196)

7.5.3 4 位全加器	(198)
习题	(199)
第 8 章 仿真	(200)
内容提要	(200)
8.1 概述	(200)
8.2 系统任务与系统函数	(201)
8.3 用户自定义元件 (UDP)	(205)
8.3.1 组合电路 UDP 元件	(206)
8.3.2 时序逻辑 UDP 元件	(208)
8.4 延时模型的表示	(210)
8.4.1 时间标尺定义 'timescale'	(210)
8.4.2 延时的表示与延时说明块	(211)
8.5 数字电路的仿真	(212)
8.5.1 测试平台 (Test Bench)	(212)
8.5.2 测试程序的编写	(213)
8.5.3 组合电路的仿真	(215)
8.5.4 时序电路的仿真	(219)
习题	(220)
第 9 章 Verilog HDL 数字电路设计实践	(221)
内容提要	(221)
9.1 基本组合电路的设计	(221)
9.1.1 门电路	(221)
9.1.2 编译码器	(223)
9.1.3 数据选择器	(226)
9.1.4 用组合电路实现的 ROM	(227)
9.2 基本时序电路的设计	(228)
9.2.1 D 触发器与 JK 触发器	(228)
9.2.2 锁存器与寄存器	(229)
9.2.3 计数器	(232)
9.2.4 ROM/RAM 模块	(233)
9.2.5 串并转换器	(234)
9.3 简易微处理器的设计	(234)
9.4 乘累加器 (MAC) 的设计	(236)
9.5 数字跑表	(238)
9.6 4 位数字频率计	(240)
9.7 交通灯控制器	(243)
9.8 乐曲演奏电路	(246)
9.8.1 音调、音长的控制	(247)
9.8.2 源代码	(248)

9.9	自动售饮料机的设计	(251)
9.10	实用多功能数字钟	(253)
9.11	计费器设计	(257)
	习题	(260)
第 10 章	设计方法与设计优化	(261)
	内容提要	(261)
10.1	设计的可综合性	(261)
10.1.1	可综合的设计	(261)
10.1.2	可综合的 Verilog HDL 结构	(262)
10.2	流水线设计技术	(264)
10.3	资源共享	(267)
10.4	有限状态机 (FSM) 设计	(271)
10.4.1	基于状态机的设计	(271)
10.4.2	频率计控制器设计举例	(273)
10.4.3	基于状态机的设计要点	(275)
10.5	多层次结构电路的设计	(277)
10.5.1	图形与文本混合设计	(277)
10.5.2	文本设计	(278)
10.6	进程	(280)
10.6.1	进程	(280)
10.6.2	进程间的通信	(281)
10.7	阻塞赋值与非阻塞赋值	(282)
10.7.1	阻塞赋值与非阻塞赋值的区别	(282)
10.7.2	使用阻塞和非阻塞赋值应遵循的原则	(283)
10.8	片内存储器的使用	(287)
10.8.1	FPGA 器件的片内存储器	(287)
10.8.2	设计举例	(288)
10.9	FPGA 设计中毛刺的消除	(290)
	习题	(293)
第 11 章	数字系统设计开发实例	(294)
	内容提要	(294)
11.1	加法器设计	(294)
11.1.1	级连加法器	(294)
11.1.2	并行加法器	(295)
11.1.3	超前进位加法器	(296)
11.1.4	流水线加法器	(299)
11.2	乘法器设计	(299)
11.2.1	并行乘法器	(299)
11.2.2	移位相加乘法器	(300)

11.2.3	查找表乘法器	(301)
11.2.4	加法树乘法器	(302)
11.3	可调信号发生器的 FPGA 实现	(304)
11.3.1	顶层设计	(304)
11.3.2	波形数据 ROM 定制	(308)
11.3.3	编译与仿真	(311)
11.3.4	引脚锁定及编程下载	(314)
11.3.5	使用嵌入式逻辑分析仪进行实时测试	(315)
11.4	设计 FIR 数字滤波器	(319)
11.4.1	FIR 滤波器的结构	(319)
11.4.2	抽头系数的编码	(320)
11.4.3	源代码及仿真	(320)
11.5	数字相关器	(323)
11.6	信道编译码器的实现	(325)
11.6.1	线性分组码编译码器	(326)
11.6.2	循环码编译码器	(328)
11.7	CRC 校验码	(332)
	习题	(334)
第 12 章	实验与设计	(335)
	内容提要	(335)
	实验 1 MAX+plus II 原理图方式设计二分频器	(335)
一、	实验目的	(335)
二、	实验步骤	(335)
	实验 2 MAX+plus II 原理图方式设计半加器	(341)
一、	实验目的	(341)
二、	实验内容	(341)
三、	实验步骤	(341)
	实验 3 Quartus II 原理图方式设计 4×4 二进制乘法器	(343)
一、	实验目的	(343)
二、	实验内容	(343)
三、	实验思考题	(347)
	实验 4 MAX+plus II 文本方式设计分频器	(347)
一、	实验目的	(347)
二、	实验步骤	(347)
三、	实验思考题	(349)
	实验 5 原理图和文本混合设计方式	(350)
一、	实验目的	(350)
二、	实验内容	(350)
	实验 6 Synplify 文本设计	(353)

一、实验目的	(353)
二、实验内容	(353)
三、实验步骤	(354)
四、实验思考题	(356)
实验 7 Verilog HDL 时序逻辑电路的设计	(356)
一、实验目的	(356)
二、实验内容	(357)
三、实验步骤	(357)
四、实验思考题	(358)
实验 8 阻塞赋值与非阻塞赋值的区别	(359)
一、实验目的	(359)
二、实验内容	(359)
三、实验步骤	(360)
四、实验注意事项	(360)
五、实验思考题	(360)
实验 9 流水线设计技术及性能分析	(361)
一、实验目的	(361)
二、实验内容	(361)
三、实验步骤	(363)
实验 10 使用有限状态机进行设计	(364)
一、实验目的	(364)
二、实验内容	(364)
三、实验步骤	(366)
四、实验注意事项	(366)
五、实验思考题	(366)
附录 A 相关术语与缩略语	(368)
附录 B Verilog HDL (IEEE Std 1364-1995) 关键字	(372)
附录 C Synplify Pro/Synplify 可综合的 Verilog 结构	(373)
附录 D MAX+plus II 支持的 Verilog HDL 结构	(374)
参考文献	(375)

第 1 章 EDA 技术与数字系统设计

内 容 提 要

- ◇ EDA 技术
- ◇ Top-down 设计和 Bottom-up 设计
- ◇ IP 复用 (IP Reuse) 技术
- ◇ 系统芯片 (SoC)
- ◇ 专用集成电路 (ASIC)
- ◇ 硬件描述语言 (HDL)
- ◇ 输入 (Design Entry)、综合 (Synthesis)
- ◇ 适配 (Fitter)、仿真 (Simulation)、编程 (Program)
- ◇ 集成的 CPLD/FPGA 开发工具
- ◇ 逻辑综合器、仿真器

本章对 EDA 技术及有关的概念进行综述, 介绍数字系统的设计技术、设计方式与实现方式等, 对数字系统的设计流程和常用的 EDA 开发工具做了阐述。

1.1 引 言

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生产制造技术和电子设计技术的进步。

生产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惊人地吻合, 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。还有一种器件的发明与使用大大改变了我们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) 等高级形式的发展。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。PLD 器件目前仍在朝密度更高、速度更快、功耗更低、功能更强的方向发展。

电子设计技术的发展也是日新月异的, 电子系统的设计理念和设计方法在过去的几十年

里发生了深刻的变化。从电子 CAD (Computer Aided Design)、电子 CAE (Computer Aided Engineering) 到电子设计自动化 (EDA, Electronic Design Automation), 设计的自动化程度越来越高, 设计的复杂性也越来越强。

目前, EDA 技术已成为电子设计技术的有力工具, 没有 EDA 技术的支持, 想要完成超大规模集成电路的设计制造是不可想像的。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。

1.2 EDA 技术及其发展

在现代的电子设计中, 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, 可以大大提高工作效率, 起到事半功倍的效果。

EDA 即电子设计自动化。EDA 技术的发展是以计算机科学、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起来的。简单地说, EDA 就是立足于计算机工作平台而开发出来的一整套先进的设计电子系统的软件工具。

一般认为, EDA 技术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。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 板的布局布线, 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。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百万、上千万乃至上亿个晶体管, 芯片的工作速度达到了 Gbps 数量级, 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, 无论设计芯片还是设计系统, 没有 EDA 工具的支持, 都是难以完成的。EDA 工具已经成为设计师必不可少的武器, 起着越来越重要的作用。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, 以及进行电子设计涉及到的各个方面。从一个角度看, EDA 技术可粗略分为系统级、门级和物理实现级 3 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 板设计到 FPGA 开发等, EDA 技术的功能和范畴如图 1.1 所示。

从发展的过程看, 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。

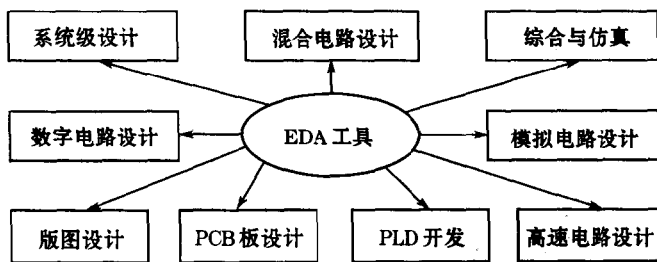


图 1.1 EDA 技术的功能和范畴

EDA 技术将向着智能性更高、功能更强、高层综合的方向发展, 另外一个发展方向就是支持软、硬件协同设计。其次, 随着芯片规模的扩大, 一些新的问题也需要 EDA 工具加以解决, 比如, 在大型的系统设计中, 设计验证工作比设计本身还要艰巨, 因此, 更好更快的仿真验证工具也是 EDA 需要加以完善的地方。EDA 的发展是没有止境的, 它将在诸多因素的推动下不断前进。

具体来说, 现代 EDA 技术具有以下一些特点。

(1) 高层综合和优化

为了能更好地支持自顶向下的设计方法, 现代的 EDA 工具能够在系统级进行综合和优化, 这样就缩短了设计的周期, 提高了设计效率。

(2) 采用硬件描述语言进行设计

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 技术的另一个特征。与传统的原理图设计方法相比, HDL 语言更适合描述规模大的数字系统, 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 语言设计的突出优点是: 语言的公开可利用性; 设计与工艺的无关性; 宽范围的描述能力; 便于组织大规模系统的设计; 便于设计的复用、交流、保存和修改等。目前最常用的硬件描述语言有 VHDL 和 Verilog HDL, 它们都已经成为 IEEE 标准。

(3) 开放性和标准化

现代 EDA 工具普遍采用标准化和开放性框架结构, 任何一个 EDA 系统只要建立了一个符合标准的开放式框架结构, 就可以接纳其他厂商的 EDA 工具一起进行设计工作。这样就可以实现各种 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。

不同的 EDA 开发工具可以组合使用, 共享设计数据, 相互间有良好的接口, 有效提高了设计者的工作效率和设计灵活性, 为用户提供了方便。

1.3 数字系统的设计技术

近 20 年来, 电子系统的设计方法和设计技术都发生了深刻的变化。在以前, 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, 用户只能根据需求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管脚, 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, 有两种基本的设计思路, 一种是自顶向下的设计思路, 一种是自底向上的设计思路。下面分别进行介绍。

1.3.1 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

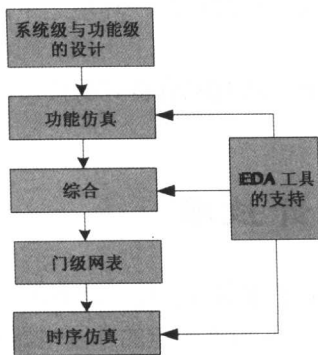


图 1.2 Top-down 设计方式

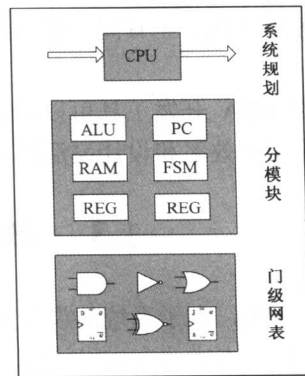


图 1.3 CPU 的 Top-down 设计方式示意图