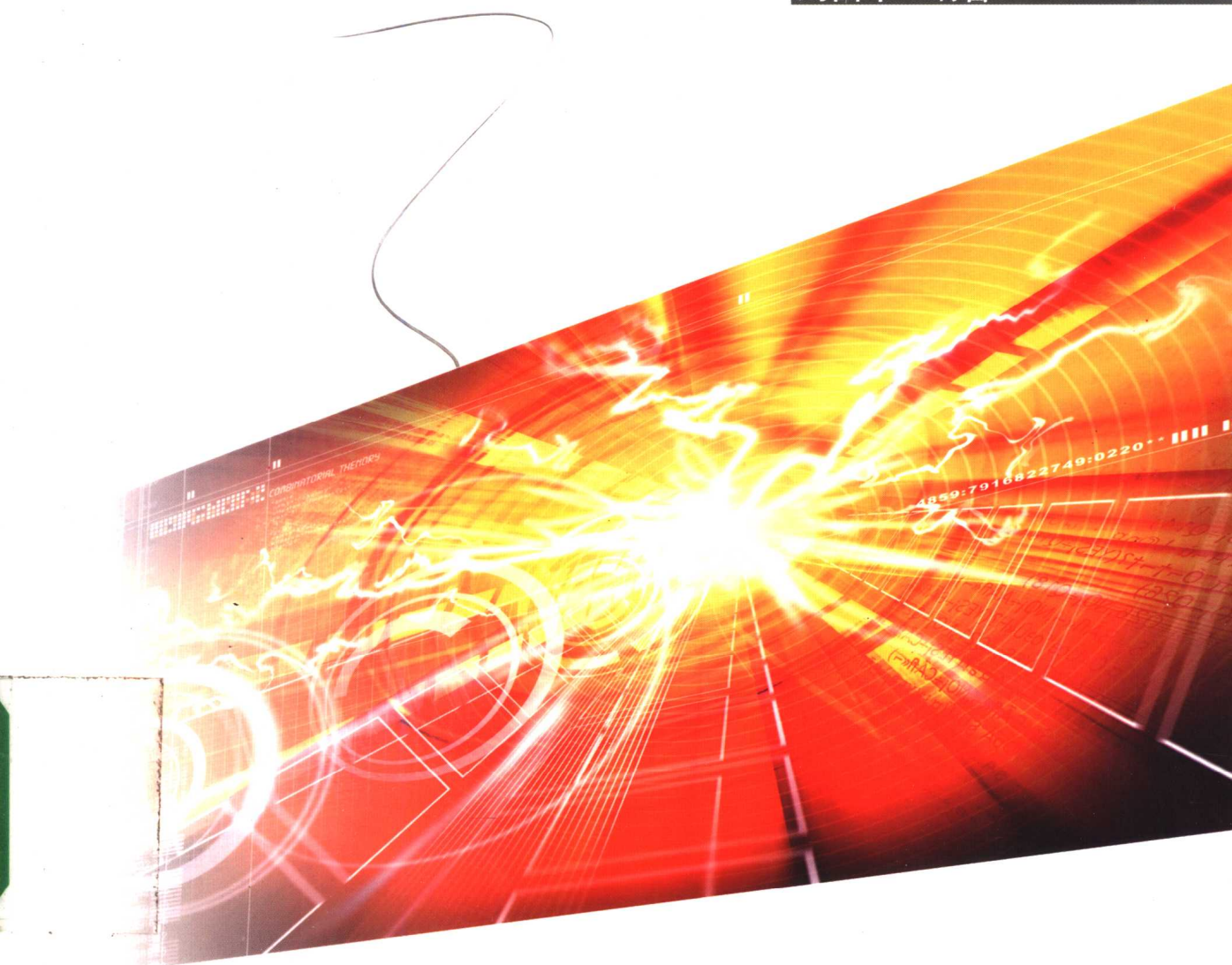


可编程逻辑器件快速进阶丛书

数字逻辑 EDA 设计与实践

——MAX + plus II 与 Quartus II 双剑合璧

刘昌华 编著



國防工業出版社

National Defense Industry Press

TN702
127

可编程逻辑器件快速进阶丛书

数字逻辑 EDA 设计与实践

……*MAX+plus II* 与 *Quartus II* 双剑合璧

刘昌华 编著

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

数字逻辑 EDA 设计与实践:MAX+plus II与 Quartus II双
剑合璧/刘昌华编著. —北京:国防工业出版社,2006.8
(可编程逻辑器件快速进阶丛书)
ISBN 7-118-04656-6

I. 数... II. 刘... III. 电子电路—电路设计:计
算机辅助设计 IV. TN702

中国版本图书馆 CIP 数据核字(2006)第 080503 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100044)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 26 字数 664 千字

2006 年 8 月第 1 版第 1 次印刷 印数 1—3500 册 定价 48.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

自 20 世纪 80 年代以来,微电子技术得到了飞速发展。在数字逻辑电路设计领域,大规模、超大规模的可编程逻辑器件及其软件开发工具相继推出,从而引起了数字逻辑电路设计的巨大变革。如 Altera 公司的 MAX + plus II 和 Quatus II, Xilinx 公司的 ISE4.1、Lattice 公司的 ispLEVER 和 Actel 公司的 Libero。由于 Altera 公司实施的高校合作计划,大多数工程研发人员在大学期间最早接触的而且使用得最多的就是 MAX + plus II,同时 MAX + plus II 的开发工具界面友好,操作方便,因此更容易被研发人员接受。MAX + plus II 和 Quartus II 都提供自身的综合工具和仿真工具,因此本书选择了 MAX + plus II 和 Quartus II。

在全面推进素质教育的形势下,为培养出面向 21 世纪的高素质大学生,现代高等学校的重点任务将是把学生的“潜力”转化为“能力”,培养学生的创新意识。因此,数字逻辑的研究和实现方法随之发生变化,从而促使数字逻辑的实验方法和实验手段也不断地更新、完善和开拓。借助 EDA 软件来进行数字逻辑电路的设计、模拟和调试,这种硬件软化的实验方法具有容易设计、容易修改和容易实现等优点,可有效提高实验效率。对于 EDA 软件的学习,只需要提供一个环境和一个指导书,剩下的就是学生自己的事了。因此如何提供一本好的指导书和教学参考资料是作者撰写此书的目的。

本书所涉及的内容包含 EDA 技术,可编程逻辑器件 CPLD/FPGA 的基本概念和基本知识以及 MAX + plus II 和 Quartus II 等 EDA 开发工具的使用方法和技巧,通过大量设计实例详细地介绍了基于 EDA 技术的层次化设计方法。

本书的特点是以数字逻辑电路和系统设计为主线,结合丰富的实例按照由浅入深的学习规律,循序渐进,逐步引入相关 EDA 技术和工具,通俗易懂,重点突出。本书适合作为 EDA 技术、数字逻辑基础设计、课程设计的教材和指导书,它可用于大学高年级学生、研究生教学及电子设计工程师技术培训,以提供和更新其采用 VHDL 语言和可编程逻辑器件的电子设计方法学方面的知识和技术内容,也可供从事数字逻辑电路和系统设计的电子工程师参考。

本书共分为 6 章。第 1 章介绍了 EDA 技术的发展,EDA 设计流程及其涉及的领域与发展趋势,互联网上的 EDA 资源;第 2 章介绍了 PROM、PLA、PAL、GAL、CPLD、FPGA 等各种可编程逻辑器件的电路结构、工作原理、使用方法和未来发展方向,并重点介绍了 Altera 公司几种典型 CPLD/FPGA 器件及其最新发展动态;第 3 章介绍了 MAX + plus II 软件的特点与使用方法,并通过具体实例,给出了如何使用 MAX + plus II 软件进行数字逻辑电路设计的 EDA 方法,并给出了相关的习题与实验供读者练习以加深理解;第 4 章以示例形式介绍了 VHDL 语言基础知识与设计方法;第 5 章介绍了 Quartus II 软件的特点与使用方法,MAX + plus II 与 Quartus II 的关系,并通过具体实例,给出了在 GW48EDA 开发平台上,使用 Quartus II 软件进行数字逻辑电路设计的 EDA 方法,并给出了相关习题与数字逻辑基础实验、设计型和研究型实验供读者练习以

加深理解;第6章介绍了数字系统的EDA层次化设计方法,并给出了其在测量仪器、自控系统、通信系统、计算机系统、雷达等领域的10个综合应用实例,选了8个数字系统设计课题供读者进行练习,或用于数字逻辑课程设计。为利于读者速查,在附录中给出了GW48EDA实验开发系统的使用说明。

本书的思路是作者从多年的“数字逻辑”课程教学及EDA工程实践中摸索出来的,本书体系也经过了4年的本专科教学实践的检验,效果较好。中国地质大学教授博士生导师王典洪博士审阅了初稿,武汉工业学院计算机与信息工程系主任管庶安教授对本书的选题和取材提出了大量建设性意见,华中师范大学刘世泽副教授审阅了部分书稿并提出了许多珍贵意见,在此对他们的帮助表示衷心的感谢。还有我的学生司群、赵庆、陈登科等参加了部分习题的编写和VHDL程序的编写调试,并为本书的文字校对、绘图做了大量工作;本书参考和引用了许多专家的著作,也得到了相关EDA供应商的大力支持,在此一并表示感谢。

本书大部分内容是在业余时间完成的,这当然要特别感谢我的家人在生活上的照顾及事业上的大力支持。

现代电子设计技术是发展的,相应的内容和方法也应不断改进,其中一定有许多问题值得深入探讨,书中也难免有错误和遗漏,恳请广大读者批评指正。本书相关问题请通过网站“www.whpu.edu.cn”或电子邮件 liuch@whpu.edu.cn 与我联系。

编著者

2005年5月20日于武汉

目 录

第 1 章 EDA 概述	1
1.1 EDA 技术及其发展	1
1.1.1 EDA 技术的发展历程	1
1.1.2 EDA 技术的主要内容	2
1.1.3 EDA 技术的发展趋势	2
1.2 硬件描述语言	3
1.2.1 ABEL-HDL	3
1.2.2 Verilog HDL	4
1.2.3 AHDL	4
1.2.4 VHDL	5
1.3 EDA 技术的层次化设计方法与流程	6
1.3.1 EDA 技术的层次化设计方法	6
1.3.2 基于 EDA 技术的数字逻辑系统设计流程	7
1.4 EDA 与传统硬件电路设计方法的比较	10
1.4.1 传统的硬件电路设计方法	10
1.4.2 EDA 技术的特点	11
1.5 EDA 技术在《数字逻辑》课程中的应用	12
1.6 EDA 软件简介	14
1.6.1 电子电路设计工具、仿真工具、PCB 设计软件	14
1.6.2 IC 设计软件	15
1.6.3 PLD 设计工具	16
1.6.4 其他软件	18
1.7 互联网上的 EDA 资源	19
1.8 习题	19
第 2 章 可编程逻辑基础	20
2.1 可编程逻辑器件的发展历程及特点	20
2.1.1 可编程逻辑器件的发展历程	20
2.1.2 可编程逻辑器件的特点	22
2.2 可编程逻辑器件的分类	23
2.2.1 按集成度分	23
2.2.2 按编程特性分	24
2.2.3 按结构分	24
2.3 简单 PLD 原理	25

2.3.1	PLD 中阵列的表示方法	25
2.3.2	PROM	27
2.3.3	PLA 器件	27
2.3.4	PAL 器件	29
2.3.5	GAL 器件	29
2.4	CPLD	30
2.4.1	CPLD 的基本结构	31
2.4.2	Altera 公司 MAX 系列 CPLD 简介	31
2.5	FPGA	35
2.5.1	FPGA 的基本结构	35
2.5.2	Altera 公司 FPGA 系列 FLEX 10K 器件的结构	36
2.5.3	嵌入阵列块(Embedded Array Block,EAB)	38
2.5.4	逻辑阵列块(Logic Array Block,LAB)	39
2.5.5	逻辑单元(Logic Element,LE)	39
2.5.6	快速通道互连	42
2.5.7	输入输出单元(IOE)	43
2.6	FPGA 与 CPLD 之比较	44
2.7	CPLD/FPGA 的设计流程	45
2.8	可编程逻辑器件的发展趋势	48
2.8.1	下一代可编程逻辑器件硬件上的 4 大发展趋势	48
2.8.2	下一代 EDA 开发软件的发展趋势	51
2.9	Altera 公司的 CPLD/FPGA 产品概述	53
2.10	习题	58

第 3 章	MAX+plus II 开发工具	59
3.1	MAX+plus II 的主要特点	59
3.2	MAX+plus II 软件设计流程	60
3.2.1	设计输入	60
3.2.2	设计处理	63
3.2.3	设计校验	65
3.2.4	器件编程	66
3.2.5	联机求助	67
3.3	MAX+plus II 在组合电路设计中的应用	67
3.3.1	建立图形设计文件	68
3.3.2	设计项目编译	70
3.3.3	设计项目校验	72
3.3.4	引脚锁定	74
3.3.5	器件编程下载与硬件测试	75
3.4	MAX+plus II 在时序逻辑电路设计中的应用	76
3.4.1	设计输入	76
3.4.2	设计项目校验	77

3.4.3	引脚锁定	77
3.4.4	器件编程下载与硬件测试	78
3.5	参数可设置 Altera 宏功能模块的应用	78
3.5.1	基于 LPM_COUNTER 的数控分频器设计	79
3.5.2	基于 LPM_ROM 的 4 位乘法器设计	81
3.5.3	基于 Altera 兆功能块的 4 位流水线加法器的设计	83
3.6	MAX+plus II 设计实例	89
3.7	习题	106
3.8	实验	108
	实验 3-1 原理图输入设计 8 位加法器	108
	实验 3-2 4-16 线译码器的 EDA 设计	108
	实验 3-3 计数器的 EDA 设计	109
	实验 3-4 原理图输入设计 M=100 十进制加法计数器	110
	实验 3-5 M 序列脉冲发生器设计	112
第 4 章	VHDL 设计基础	113
4.1	VHDL 的基本组成	113
4.1.1	实体	113
4.1.2	构造体	116
4.1.3	程序包	119
4.1.4	库	120
4.1.5	配置	121
4.2	VHDL 语言的基本要素	124
4.2.1	VHDL 语言的标识符(Identifiers)	124
4.2.2	VHDL 语言的客体(Object)	125
4.2.3	VHDL 语言的数据类型	127
4.2.4	VHDL 语言的运算操作符	132
4.3	VHDL 语言的基本语句	135
4.3.1	顺序描述语句	135
4.3.2	并行语句	144
4.4	常见组合逻辑电路的 VHDL 设计	154
4.4.1	基本门电路设计	154
4.4.2	编码器、译码器、选择器	156
4.4.3	加法器	160
4.4.4	数值比较器	162
4.4.5	算术逻辑运算器(ALU)	163
4.5	常见时序逻辑电路的 VHDL 设计	165
4.5.1	触发器	166
4.5.2	锁存器和寄存器	168
4.5.3	计数器	171
4.6	基于 VHDL 的设计方法综合举例	176

4.6.1	移位相加 8 位乘法器的 VHDL 设计	176
4.6.2	序列计数器的设计	182
4.6.3	简易数字钟的设计	186
4.7	习题	193
4.8	实验	197
实验 4-1	简单组合电路设计	197
实验 4-2	简单时序电路设计	197
实验 4-3	设计含计数使能、异步复位和计数值并行预置功能的 4 位加法 计数器	198
实验 4-4	设计移位运算器	199
实验 4-5	循环冗余校验(CRC)模块设计	202
第 5 章	Quartus II 开发系统	206
5.1	Quartus II 简介	206
5.1.1	Quartus II 的特点	206
5.1.2	Quartus II 系统安装要求	212
5.1.3	Quartus II 系统安装许可与技术支持	212
5.1.4	Quartus II 设计流程	213
5.2	Quartus II 设计入门	225
5.2.1	Quartus II 的启动	225
5.2.2	设计输入	229
5.2.3	编译综合	233
5.2.4	仿真测试	233
5.2.5	硬件测试	236
5.3	Quartus II 设计技巧	238
5.3.1	基于原理图输入的数字逻辑电路的 Quartus II 设计	238
5.3.2	基于 VHDL 文本输入的数字逻辑电路的 Quartus II 设计	254
5.3.3	基于 LPM 可定制宏功能模块的数字逻辑电路的 Quartus II 设计	267
5.3.4	基于混合输入方式的数字逻辑电路的 Quartus II 设计	276
5.4	从 MAX+plus II 向 Quartus II 转换	281
5.4.1	MAX+plus II 与 Quartus II 的功能比较	281
5.4.2	转换 MAX+plus II 设计	282
5.5	习题	286
5.6	实验	287
实验 5-1	Quartus II 原理图输入设计法	287
实验 5-2	Quartus II 的 VHDL 文本输入设计法	288
实验 5-3	8 位流水加法器的 EDA 设计	290
实验 5-4	Quartus II 设计正弦信号发生器	291
实验 5-5	用 EP1C6Q240C8 器件设计一个 4 位十进制数字显示的频率计电路	293
第 6 章	数字系统的 EDA 设计	296

6.1	数字系统的 EDA 层次化设计方法	297
6.1.1	设计的层次	297
6.1.2	自顶向下方法的含义	299
6.1.3	自底向上方法的含义	300
6.2	简易数字钟的 EDA 设计	301
6.2.1	设计要求	301
6.2.2	功能描述	301
6.2.3	数字钟的层次化设计方案	301
6.2.4	数字钟的顶层设计和仿真	307
6.3	数字式频率计的 EDA 设计	308
6.3.1	设计要求	308
6.3.2	原理描述	309
6.3.3	频率计的层次化设计方案	309
6.3.4	频率计电路的顶层设计和仿真	317
6.4	乐曲硬件演奏电路 EDA 设计	318
6.4.1	设计要求	318
6.4.2	原理描述	319
6.4.3	乐曲硬件演奏电路的层次化设计方案	320
6.4.4	乐曲硬件演奏电路的顶层设计和仿真	326
6.5	基于流水线技术的数字相关器 EDA 设计	328
6.5.1	设计要求	328
6.5.2	原理描述	328
6.5.3	数字相关器电路的层次化设计方案	329
6.5.4	编译设计和时序仿真	331
6.5.5	性能分析	331
6.6	红绿灯交通控制器 EDA 设计	332
6.6.1	设计要求	332
6.6.2	系统组成	332
6.6.3	红绿灯交通控制器的层次化设计方案	333
6.6.4	红绿灯交通控制器顶层电路的设计和仿真	344
6.7	出租车自动计费器 EDA 设计	347
6.7.1	设计要求	347
6.7.2	原理描述	347
6.7.3	出租车自动计费器的层次化设计方案	347
6.7.4	出租车自动计费器顶层电路的设计和仿真	354
6.8	步进电机定位控制器 EDA 设计	355
6.8.1	设计要求	355
6.8.2	原理描述	355
6.8.3	步进电机定位控制器的层次化设计方案	356
6.9	多功能算术逻辑运算单元的 EDA 设计	365
6.9.1	设计要求	365

6.9.2	原理描述	366
6.9.3	多功能算术逻辑运算单元的层次化设计方案	366
6.9.4	多功能算术逻辑运算单元的顶层设计和仿真	368
6.10	微程序控制器的 EDA 设计	369
6.10.1	设计要求	369
6.10.2	原理描述	370
6.10.3	微程序控制器的层次化设计方案	372
6.10.4	微程序控制器的顶层电路设计和仿真	376
6.11	MTI 雷达三脉冲可变结构的数字对消器的 EDA 设计	377
6.11.1	设计要求	377
6.11.2	原理描述	378
6.11.3	三脉冲可变结构对消器层次化设计方案	379
6.11.4	三脉冲可变结构对消器的顶层电路设计与仿真	383
6.12	数字系统设计课题选编	383
6.12.1	数字系统设计课题一 运算器设计	383
6.12.2	数字系统设计课题二 时序发生器设计	385
6.12.3	数字系统设计课题三 设计一个具有 3 种信号灯的 traffic 灯控制系统	387
6.12.4	数字系统设计课题四 设计一个基于 CPLD 芯片的弹道计时器	389
6.12.5	数字系统设计课题五 设计一个基于 CPLD 芯片的汽车尾灯控制器	391
6.12.6	数字系统设计课题六 数字锁设计	393
6.12.7	数字系统设计课题七 电梯控制器设计	395
6.12.8	数字系统设计课题八 自动售饮料控制器设计	399
参考文献		404

第 1 章 EDA 概述

1.1 EDA 技术及其发展

20 世纪后半期,随着集成电路和计算机的不断发展,电子技术面临着严峻的挑战。由于电子技术发展周期不断缩短,专用集成电路(ASIC)的设计面临着难度不断提高与设计周期不断缩短的矛盾。为了解决这个问题,必须采用新的设计方法和使用高层次的设计工具。在此情况下,EDA(Electronic Design Automation, 电子设计自动化)技术应运而生。

1.1.1 EDA 技术的发展历程

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,以硬件描述语言为设计语言,以可编程器件为实验载体,以 ASIC、SOC 芯片为目标器件,以数字逻辑系统设计为应用方向的电子产品自动化设计过程。

随着现代半导体的精密加工技术发展到了深亚微米($0.18\mu\text{m} \sim 0.35\mu\text{m}$)阶段,基于大规模或超大规模集成电路技术的定制或半定制 ASIC(Application Specific IC, 专用集成电路)器件大量涌现并获得广泛的应用,使整个电子技术与产品的面貌发生了深刻的变化,极大地推动了社会信息化的发展进程。而支撑这一发展进程的主要基础之一就是 EDA 技术。

EDA 技术在硬件方面融合了大规模集成电路制造技术,IC 版图设计技术、ASIC 测试和封装技术、CPLD/FPGA 技术等;在计算机辅助工程方面融合了计算机辅助设计 CAD、计算机辅助制造 CAM、计算机辅助测试 CAT 技术及多种计算机语言的设计概念;而在现代电子学方面则融合了更多的内容,如数字电路设计理论、数字信号处理技术、系统建模和优化技术等。因此 EDA 技术为现代数字系统理论和设计的表达与应用提供了可能性,它已不是某一学科的分支,而是一门综合性学科。EDA 技术打破了计算机软件与硬件间的壁垒,是计算机软件技术与硬件实现、设计效率和产品性能的合二为一,它代表了数字电子设计技术和应用技术的发展方向。

EDA 技术伴随着计算机、集成电路、电子系统设计的发展经历了 3 个发展阶段。

(1) CAD(Computer Aided Design)阶段

20 世纪 70 年代发展起来的 CAD 阶段是 EDA 技术发展的早期阶段,这一阶段集成电路制作方面, MOS 工艺得到广泛应用,可编程逻辑技术及其器件已经问世,计算机作为一种运算工具已在科研领域得到广泛应用,人们借助于计算机进行电路图的输入、存储及 PCB 版图设计,从而摆脱了用手工进行电子设计时的大量繁琐、重复、单调的计算与绘图工作,并逐步利用计算机进行电子系统的设计、分析与仿真。

(2) 电子设计 CAE(Computer Aided Engineering)阶段

CAE 即计算机辅助工程是在 20 世纪 80 年代开始应用,在 CAD 工具逐步完善的基础上发展起来的。此时集成电路设计技术进入了 CMOS(互补场效应管)时代,复杂可编程逻辑器件已进入商业应用领域,相应的辅助设计软件也已投入使用。

在这一阶段,人们已将各种电子线路设计工具如电路图输入、编译与连接、逻辑模拟、仿真分析、版图自动生成及各种单元库都集成在一个 CAE 系统中,以实现电子系统或芯片从原理图输入到

版图设计输出的全程设计自动化。利用现代的 CAE 系统,设计人员在进行系统设计的时候,已可以把反映系统互连线路对系统性能的影响因素,如板级电磁兼容、板级引线走向等一并考虑进去,使电子系统的设计与开发工作更贴近产品实际,更加自动化、更加方便和稳定可靠,大大提高了工作效率。

(3) EDA 阶段

20 世纪 90 年代后期,出现了以硬件描述语言、系统级仿真和综合技术为特征的 EDA 技术。随着硬件描述语言 HDL 的标准得到进一步的确立,计算机辅助工程、辅助分析、辅助设计在电子技术领域获得更加广泛的应用,与此同时电子技术在通信、计算机及家电产品生产中的市场和技术需求,极大推动了全新的电子自动化技术的应用和发展。特别是集成电路设计工艺步入了深亚微米阶段,百万门以上的大规模可编程逻辑器件的陆续面世,以及基于计算机技术的面向用户的低成本大规模 ASIC 设计技术的应用,促进了 EDA 技术的形成。在这一阶段,电路设计者只需要完成对系统功能的描述,就可以由计算机软件进行系列处理,最后得到设计结果,并且修改设计方案如同修改软件一样方便,利用 EDA 工具可以极大地提高设计效率。

这时的 EDA 工具不仅具有电子系统设计的能力,而且能提供独立于工艺和厂家的系统级设计能力,具有高级抽象的设计构思手段。因此,可以说 20 世纪 90 年代的 EDA 技术是电子电路设计的革命。

1.1.2 EDA 技术的主要内容

根据 1.1.1 节所述,EDA 技术是以大规模可编程逻辑器件为设计载体,以硬件描述语言为系统逻辑描述的主要表达方式,以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具,通过有关的开发软件,自动完成用软件的方法设计电子系统到硬件系统的逻辑编译、逻辑化简、逻辑分割、逻辑综合及优化、逻辑布局布线、逻辑仿真,直至对于特定目标芯片的适配编译、逻辑映射、编程下载等工作,最终形成集成电子系统或专用集成芯片的一门新技术。

EDA 技术涉及面广,内容丰富,从教学和实用的角度看,主要有以下 4 个方面内容:①大规模可编程逻辑器件;②硬件描述语言;③软件开发工具;④实验开发系统。其中,大规模可编程逻辑器件是利用 EDA 技术进行电子系统设计的载体;硬件描述语言是利用 EDA 技术进行电子系统设计的主要表达手段;软件开发工具是利用 EDA 技术进行电子系统设计的智能化、自动化设计工具;实验开发系统是利用 EDA 技术进行电子系统设计的下载工具及硬件验证工具。

利用 EDA 技术进行数字逻辑系统设计,具有以下特点。

- ① 全程自动化:用软件方式设计的系统到硬件系统的转换,是由有关的开发软件自动完成的。
- ② 工具集成化:具有开放式的设计环境,这种环境也称为框架结构(Framework),它在 EDA 系统中负责协调设计过程和管理设计数据,实现数据与工具的双向流动。它的优点是可以将不同公司的软件工具集成到一个统一的计算机平台上,使之成为一个完整的 EDA 系统。
- ③ 操作智能化:使设计人员不必学习许多深入的专业知识,也可免除许多推导运算即可获得优化的设计成果。
- ④ 执行并行化:由于多种工具采用了统一的数据库,使得一个软件的执行结果马上可被另一个软件使用,使得原来要串行的设计步骤变成了并行过程,也称为“同时工程(Concurrent Engineering)”。
- ⑤ 成果规范化:都采用 VHDL(超高速集成电路硬件描述语言),它是 EDA 系统的一种输入模式,可以支持从数字系统级到门级的多层次的硬件描述。

1.1.3 EDA 技术的发展趋势

EDA 技术在进入 21 世纪后得到了更大的发展,突出表现在以下几个方面。

- ① 使电子设计成果以自主知识产权的方式明确表达和确认成为可能。

② 使仿真和设计两方面支持标准硬件描述语言、功能强大的 EDA 软件不断推出。

③ 电子技术全方位纳入 EDA 领域,除了日益成熟的数字技术外,传统的电路系统设计建模理念发生了重大的变化:软件无线电技术的崛起,模拟电路系统硬件描述语言的表达和设计的标准化,系统可编程模拟器件的出现,数字信号处理和图像处理的全硬件实现方案的普遍接受,软、硬件技术的进一步融合等。

④ EDA 使得电子领域各学科的界限更加模糊,更加互为包容:模拟与数字、软件与硬件、系统与器件、专用集成电路 ASIC 与 FPGA(Field Programmable Gate Array)、行为与结构等的界限更加模糊,更加互为包容。

⑤ 更大规模的 FPGA 和 CPLD(Complex Programmable Logic Device)器件的不断推出。

⑥ 基于 EDA 工具的 ASIC 设计标准单元已涵盖大规模电子系统及 IP 核模块。

⑦ 软件 IP 核在电子行业的产业领域、技术领域和设计应用领域得到进一步确认。

⑧ 单片电子系统 SOC(System On Chip)高效、低成本设计技术的成熟。

总之,随着系统开发对 EDA 技术的目标器件的各种性能要求的提高,ASIC 和 FPGA 将更大程度相互融合。这是因为虽然标准逻辑器件 ASIC 芯片尺寸小、功能强大、耗电小,但设计复杂,并且有批量生产要求;可编程逻辑器件开发费用低廉,能在现场进行编程,但体积大、功能有限,而且功耗较大。因此,FPGA 和 ASIC 互相融合,取长补短。由于一些 ASIC 制造商提供具有可编程逻辑的标准单元,可编程器件制造商重新对标准逻辑单元发生兴趣,而有些公司采取两头并进的方法,从而使市场开始发生变化,在 FPGA 和 ASIC 之间正在诞生一种“杂交”产品,以满足成本和上市速度的要求,例如将可编程逻辑器件嵌入标准单元。

现今也在进行将 ASIC 嵌入可编程逻辑单元的工作。目前,许多 PLD 公司开始为 ASIC 提供 FPGA 内核,PLD 厂商与 ASIC 制造商结盟,为 SOC 设计提供嵌入式 FPGA 模块,使未来的 ASIC 供应商有机会更快地进入市场,利用嵌入式内核获得更长的市场生命期。传统 ASIC 和 FPGA 之间的界限正变得模糊。系统级芯片不仅集成 RAM 和微处理器,也集成 FPGA,整个 EDA 和 IC 设计工业都朝这个方向发展。

1.2 硬件描述语言

硬件描述语言(Hardware Description Language, HDL)是电子系统硬件行为描述、结构描述、数据流描述的语言。目前利用硬件描述语言可以进行数字电子系统的设计。随着研究的深入,利用硬件描述语言进行模拟电子系统设计或混合电子系统设计,也正在探索中。

硬件描述的语言种类很多,有的从 Pascal 发展而来,也有一些从 C 语言发展而来。有些 HDL 成为 IEEE 标准,但大部分是企业标准。HDL 发展的技术源头是:在 HDL 形成之前,已有了许多程序设计语言,如汇编、C、Pascal、Fortran、Prolog 等。这些语言运行在不同硬件平台、不同的操作环境中,它们适合于描述过程和算法,不适合做硬件描述。CAD 的出现,使人们可以利用计算机进行建筑、服装等行业的辅助设计,而电子辅助设计也同步发展起来。在利用 EDA 工具进行电子设计时,逻辑图、分立电子元件作为整个越来越复杂的电子系统的设计工具已不适应。任何一种 EDA 工具,都需要一种硬件描述语言来作为 EDA 工具的工作语言。这些众多的 EDA 工具软件开发者,各自推出了自己的 HDL 语言。在我国比较有影响的硬件描述语言有:ABEL-HDL 语言、Verilog HDL 语言、AHDL 语言和 VHDL 语言。

1.2.1 ABEL-HDL

这是一种早期的硬件描述语言。在可编程逻辑器件的设计中,可方便准确地描述所设计的电路

逻辑功能。它支持逻辑电路的多种表达形式,其中包括逻辑方程、真值表和状态图。ABEL 语言和 Verilog 语言同属一种描述级别,但 ABEL 语言的特性受支持的程度远远不如 Verilog。Verilog 是从集成电路设计中发展而来,语言较为成熟,支持的 EDA 工具很多。而 ABEL 语言从早期可编程逻辑器件(PLD)的设计中发展而来。ABEL-HDL 被广泛用于各种可编程逻辑器件的逻辑功能设计,由于其语言描述的独立性,因而适用于各种不同规模的可编程器的设计。如 DOS 版的 ABEL 3.0 软件可对包括 GAL 器件进行全方位的逻辑描述和设计,而在诸如 Lattice 的 ispExpert、Data I/O 的 Synario、Vantis 的 Design-Direct、Xilinx 的 Foundation 和 webPACK 等 EDA 软件中,ABEL-HDL 同样可用于较大规模的 FPGA/CPLD 器件的功能设计。ABEL-HDL 还能对所设计的逻辑系统进行功能仿真。ABEL-HDL 的设计也能通过标准格式设计转换文件,转换成其他设计环境,如 VHDL、Verilog HDL 等。从长远来看,VHDL 和 Verilog HDL 的应用会比 ABEL-HDL 广泛,ABEL-HDL 只会在较小的范围内继续存在。

1.2.2 Verilog HDL

Verilog HDL 就是在应用最广泛的 C 语言的基础上发展起来的一种硬件描述语言,它是由 GDA(Gateway Design Automation)公司的 Phil Moorby 在 1983 年末首创的,最初只设计了一个仿真与验证工具,之后又陆续开发了相关的故障模拟与时序分析工具。1985 年 Moorby 推出它的第 3 个商用仿真器 Verilog-XL,获得了巨大的成功,从而使得 Verilog HDL 迅速得到推广应用。1989 年 Cadence 公司收购了 GDA 公司,使得 Verilog HDL 成为了该公司的独家专利。1990 年 Cadence 公司公开发表了 Verilog HDL,并成立 LVI 组织以促进 Verilog HDL 成为 IEEE 标准,即 IEEE Standard 1364—1995。

Verilog HDL 的最大特点就是易学易用,如果有 C 语言的编程经验,可以在一个较短的时间内很快地学习和掌握,因而可以把 Verilog HDL 内容安排在与 ASIC 设计等相关的课程内部进行讲授,由于 HDL 语言本身是专门面向硬件与系统设计的,这样的安排可以使学习者同时获得设计实际电路的经验。与之相比,VHDL 的学习要困难一些。但 Verilog HDL 较自由的语法也容易造成初学者犯一些错误,这一点要特别注意。

1.2.3 AHDL

Altera 硬件描述语言(Altera Hardware Description Language, AHDL)是一种模块化的高级语言,是 Altera 公司发明的 HDL,特点是非常易学易用,学过高级语言的人可以在很短的时间(如几周)内掌握 AHDL。

AHDL 语言完全集成于 MAX+plus II/Quartus II 系统之中,特别适于描述复杂的组合逻辑、组(Group)运算、状态机、真值表和参数化逻辑。设计者可以通过 MAX+plus II 或 Quartus II 的软件系统对 AHDL 源程序进行编译,并通过对源程序的编译建立仿真、时域分析和器件编程的输出文件。

AHDL 的语句和元素种类齐全、功能强大,而且易于使用。设计者可以使用 AHDL 建立完整层次的工程(Project)设计文件,或者在一个层次的设计中混合其他类型的设计文件,如 Verilog HDL 或 VHDL 设计文件。AHDL 文件作为一种文本文件,它既可以用 EDA 提供的文本文件编辑器来建立文本(AHDL Text Design File, TDF),也可以用其他文本编辑器建立文本文件,但是,由于 AHDL 与 MAX+plus II/Quartus II 间的特殊关系,建议最好使用前者,设计者可以方便地运用 MAX+plus II/Quartus II 对 AHDL 进行文本编辑、编译、调试等工作,尤其是在消息处理器中对错误自动定位的功能使调试十分方便。它的缺点是移植性不好,通常只用于 Altera 公司自己的开发系统。

1.2.4 VHDL

VHDL(Very High Speed Integrated Circuit Hardware Description Language)即超高速集成电路硬件描述语言,美国国防部在20世纪80年代后期开发了VHDL语言。VHDL工作小组于1981年6月成立,提出了一个满足电子设计各种要求的能够作为工业标准的HDL。1983年第3季度,由IBM公司、TI公司、Intermetrics公司签约,组成开发小组,工作任务是提出语言版本和开发软件环境。1986年IEEE标准化组织开始工作,讨论VHDL语言标准,历时一年有余;1987年12月通过标准审查,并宣布实施,即IEEE STD 1076—1987[LRM87]。1993年VHDL重新修订,形成新的标准,即IEEE STD 1076—1993[LRM93]。

该语言设计技术齐全、方法灵活、可与制作工艺无关、编程易于共享,所以成为硬件描述语言的主流,成为标准硬件描述语言。将VHDL程序写入可编程芯片中,做成ASIC芯片,因其开发周期短,更改方便,所以将在很大范围内取代单片控制电路,成为未来数字电路设计的主流。由于半导体编程技术的快速进步,VHDL所能提供的高阶电路描述语言方式,使复杂的电路可以通过VHDL编辑器的电路方式,轻易而且快速地达到设计的规格。VHDL电路描述语言所能涵盖的范围相当广,能适用于各种不同阶层的设计工程师的需求。从ASIC的设计到PCB系统的设计,VHDL电路描述语言都能派上用场,所以VHDL电路设计毫无疑问地成为硬件设计工程师的必备工具。目前,VHDL语言也已成为FPGA/CPLD编程最常用的工具。

VHDL作为EDA的重要组成部分,提供了借助计算机进行数字系统设计的一种很好的手段。用VHDL进行设计有许多优点,VHDL的硬件描述能力很强,可以用于从门级、电路级直至系统级的描述、仿真、综合和调试。利用VHDL丰富的仿真语句和库函数,对大系统的早期设计,可在远离门级的高层次上进行模拟,以利于设计者确定整个设计结构和功能的可行性。VHDL强大的行为描述能力和程序结构,使其具有支持对大规模设计进行分解,以及对已有的设计进行再利用的功能。运用VHDL设计系统硬件具有相对独立性,设计时没有嵌入与工艺有关的信息,对硬件的描述与具体的工艺技术和硬件结构无关。当门级或门级以上的描述通过仿真检验以后,再用相应的工具将设计映射成不同的工艺,这使硬件实现的目标器件有很宽的选择范围,并且修改电路与修改工艺(或选择器件)相互之间不会产生不良影响。VHDL标准、规范,语法较为严格,采用VHDL的设计便于重复利用和交流,VHDL所具有的类属描述语句和子程序调用等功能,使设计者对完成的设计不必改变源程序,只需改变类属参数或函数,就可改变设计的规模和结构。当然,VHDL也存在某些不足,如电路采用高级而简明的文本文件方式进行描述的同时,放弃了对电路门级实现定义的控制;由综合工具进行逻辑综合的实现效果有时不太理想;工具的不同也导致了综合质量的差异。

由于VHDL是一种硬件描述语言,设计者需要较多地了解数字电路与逻辑设计方面的电路知识;而更为重要的是必须以一种并行语言的思路去理解和应用VHDL。VHDL描述的是实际的电路系统。电路系统内各功能模块的工作状态可以相互独立、互不相关,也可以互为因果;也就是说,任一时刻,电路系统中可有許多相关的和不相关的事件同时发生,为适应实际电路系统的工作方式,VHDL以顺序和并行的多种语句的方式对同一时刻电路中所有可能发生的事件进行描述。可以认为,VHDL是一种并行语言。当然,VHDL仍保留着计算机语言的基本特征。用VHDL进行的设计描述只是综合器赖以构建硬件电路结构的依据,不可能代表硬件电路真实的行为方式。如用VHDL的顺序语句可以描述多路选择器、译码器等以并行工作为特征的电路,但实际电路并不按照逐个顺序判断的工作方式进行。因而,尽可能了解软件语句与硬件结构间的联系,了解软件描述背后的硬件工作行为和硬件结构方式,将有助于实现高质量的VHDL设计。

总之,VHDL是EDA技术最重要的应用工具。离开硬件描述语言的支持,EDA技术将无法应用。1995年我国国家技术监督局制定的《CAD通用技术规范》推荐将VHDL作为我国电子设计自

动化硬件描述语言的国家标准。目前, VHDL 已经成为世界上各家 EDA 工具和集成电路厂商普遍认同和共同推广的标准化硬件描述语言。掌握 VHDL, 利用 VHDL 设计电子电路, 是当前进行技术竞争的一项基本技能和强有力工具。

1.3 EDA 技术的层次化设计与流程

EDA 技术的出现使数字系统的分析与设计方法发生了根本的变化, 采用的基本设计方法主要有 3 种: 直接设计、自顶向下(Top-Down)设计、自底向上(Bottom-Up)设计。直接设计就是将设计看成一个整体, 将其设计成为一个单电路模块, 它适合小型简单的系统。而一些功能较复杂的大型数字逻辑系统设计适合采用自顶向下或自底向上的设计方法。自顶向下的设计方法就是从设计的总体要求入手, 自顶向下地将设计划分为不同的功能子模块, 每个模块完成特定的功能, 这种设计方法首先确定顶层模块的设计, 再进行子模块的详细设计, 而在子模块的设计中可以调用库中已有的模块或设计过程中保留下来的实例。自底向上的设计方法与自顶向下的设计方法恰恰相反。

1.3.1 EDA 技术的层次化设计方法

在 EDA 设计中往往采用层次化的设计方法, 分模块、分层次地进行设计描述。描述系统总功能的设计为顶层设计, 描述系统中较小单元的设计为底层设计。整个设计过程可理解为从硬件的顶层抽象描述向最底层结构描述的一系列转换过程, 直到最后得到可实现的硬件单元描述为止。层次化设计方法比较自由, 既可采用自顶向下(Top-Down)设计也可采用自底向上(Bottom-Up)设计, 可在任何层次使用原理图输入和硬件描述语言 HDL 设计。

(1) 自底向上(Bottom-Up)设计方法

Bottom-Up 设计方法的中心思想是首先根据对整个系统的测试与分析, 由各个功能块连成一个完整的系统, 由逻辑单元组成各个独立的功能模块, 由基本门构成各个组合与时序逻辑单元。

Bottom-Up 设计方法的特点: 从底层逻辑库中直接调用逻辑门单元; 符合硬件工程师传统的设计习惯; 在进行底层设计时缺乏对整个电子系统总体性能的把握; 在整个系统完成后, 要进行修改较为困难, 设计周期较长; 随着设计规模与系统复杂度的提高, 这种方法的缺点更突出。

传统的数字系统的设计方法一般都是自底向上的, 即首先确定构成系统的最底层的电路模块或元件的结构和功能, 然后根据主系统的功能要求, 将它们组成更大的功能块, 使它们的结构和功能满足高层系统的要求, 依此类推, 直至完成整个目标系统的 EDA 设计。

例如, 对于一般数字系统的设计, 使用自底向上的设计方法, 必须首先确定使用的器件类别, 如 74 系列的器件、某种 RAM 和 ROM、某类 CPU 以及某些专用功能芯片等, 然后是构成多个功能模块, 如数据采集、信号处理、数据交换和接口模块等, 直至最后利用它们完成整个系统的设计。

(2) 自顶向下(Top-Down)设计方法

Top-Down 设计方法的中心思想是: 系统层是一个包含输入输出的顶层模块, 并用系统级、行为描述加以表达, 同时完成整个系统的模拟和性能分析; 整个系统进一步由各个功能模块组成, 每个模块由更细化的行为描述加以表达; 由 EDA 综合工具完成到工艺库的映射。

Top-Down 设计方法的特点: 结合模拟手段, 可以从开始就掌握目标系统的性能状况; 随着设计层次向下进行, 系统的性能参数将进一步得到细化与确认; 可以根据需要及时调整相关的参数, 从而保证了设计结果的正确性, 缩短了设计周期; 当规模越大时, 这种方法的优越性越明显; 须依赖 EDA 设计工具的支持及昂贵的基础投入; 逻辑综合及以后的设计过程的实现, 均需要精确的工艺库的支持。

现代数字系统的设计方法一般都是自顶向下的层次化设计方法, 即从整个系统的整体要求出发,