

EDA

技术实用教程

孟庆辉 刘辉 程继航 石静苑 © 编著



国防工业出版社

National Defense Industry Press

TN702
342
12

内容简介

EDA 技术实用教程

孟庆辉 刘辉 程继航 石静苑 编著

图书在版编目(CIP)数据

EDA技术实用教程 / 孟庆辉, 刘辉, 程继航, 石静苑编著. — 北京: 国防工业出版社, 2008.9

ISBN 978-7-118-03212-2

I. E... II. 孟... III. 电子工业—教材 IV. TN702

中国版本图书馆CIP数据核字(2008)第094434号

本书由国防工业出版社出版

国防工业出版社

(北京市东城区东直门南大街2号 邮编: 100044)

发行部电话: 010-61512754

发行部传真: 010-61512754

本书 2008年9月第1版第1次印刷 印张: 16.5 字数: 400千字 定价: 39.00元

国防工业出版社

发行部电话: (010) 68438433 发行部传真: (010) 68411333
发行部地址: 北京·东直门南大街2号

内 容 简 介

全书共分9章,主要讲解EDA工具软件Quartus II 5.0的操作,硬件描述语言VHDL及其应用实例,可编程逻辑器件的结构组成和工作原理,GW48EDA/SOPC实验开发系统的使用方法。全书理论与实践操作紧密结合,体系合理、内容全面、概念清晰、通俗易懂,有助于读者系统学习和掌握所学知识。

本书可作为本科生教材,也适合初学者培训使用。

图书在版编目(CIP)数据

EDA技术实用教程/孟庆辉等编著. —北京:国防工业出版社,2008.9

ISBN 978-7-118-05821-5

I. E... II. 孟... III. 电子电路—电路设计:计算机辅助设计—教材 IV. TN702

中国版本图书馆CIP数据核字(2008)第094454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 21¼ 字数 490千字

2008年9月第1版第1次印刷 印数 1—4000册 定价 39.00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

电子设计自动化(Electronic Design Automation, EDA)是20世纪90年代初以来迅速发展起来的现代电子工程领域的一门新技术。EDA技术是以计算机科学和微电子技术发展为先导,汇集了计算机图形学、拓扑逻辑学、微电子工艺与结构学和计算数学等多种计算机应用学科最新成果的先进技术。它以可编程逻辑器件(PLD)为载体,以计算机为工作平台,以EDA工具软件为开发环境,以硬件描述语言(HDL)为电子系统功能描述方式,以电子系统设计为应用方向的电子产品自动化设计过程。

本书是根据EDA包含的主要内容编写的,既考虑了EDA技术本身的系统性和完整性,又考虑了EDA技术的实用性和实践性,做到理论与实践的有机结合。无论是器件、硬件描述语言的讲解,还是EDA工具软件、EDA应用系统介绍,都是以当前应用广泛的主流内容来展开,力求做到内容新颖、重点突出、讲解精炼、强化实践。

本书共有9章。第1章介绍EDA技术的发展、基本设计工具、设计流程和发展趋势。第2、3章为EDA技术的软件操作。主要介绍了FPGA/CPLD的开发流程及工具中各功能模块的功能。开发操作环境主要介绍Altera公司Quartus II 5.0的主要功能,硬件语言主要介绍VHDL语言及其应用实例。第4、5、7、9章为设计实例和多项实验,读者可通过实验操作很好地掌握EDA的开发设计方法和Quartus II 5.0等工具软件的使用技能,提高EDA技术应用和实践能力。第6章介绍可编程逻辑器件的结构组成、工作原理,特别是世界上相关主流公司(如Altera公司、Xilinx公司、Lattice公司)的可编程逻辑器件的结构组成、特点及其性能指标。第8章介绍了由杭州康芯电子有限公司研制开发的、系统性能相对较好的GW48EDA/SOPC实验开发系统的使用方法,以使读者能具体地了解基于某种EDA平台的VHDL逻辑设计所必需的硬件仿真和实验验证的方法与过程。

本书由孟庆辉、刘辉、程继航、石静苑编著,焦阳、杨庆禹、裘昌利、王兆新、甘世春参加编写,董兴文主审。由于作者水平有限,书中难免存在不妥之处,敬请读者批评指正。

编者

目 录

第1章 EDA 技术概述	1
1.1 EDA 技术及其发展	1
1.1.1 EDA 技术的涵义	1
1.1.2 EDA 技术的发展史	1
1.2 EDA 技术的基本工具	4
1.2.1 设计输入编辑器	4
1.2.2 仿真器	4
1.2.3 HDL 综合器	5
1.2.4 适配器(布局、布线器)	5
1.2.5 下载器(编程器)	6
1.3 EDA 技术的基本设计思路	6
1.3.1 EDA 电路级设计	6
1.3.2 EDA 系统级设计	6
1.4 EDA 的设计流程	7
1.4.1 设计准备	7
1.4.2 设计输入	8
1.4.3 设计处理	8
1.4.4 设计检验	9
1.4.5 器件编程与配置	9
1.5 EDA 的发展趋势	10
1.5.1 可编程器件的发展趋势	10
1.5.2 软件开发工具的发展趋势	10
1.5.3 输入方式的发展方向	11
第2章 Quartus II 5.0 设计软件	13
2.1 概述	13
2.2 Quartus II 5.0 软件安装	13
2.2.1 系统配置要求	13
2.2.2 Quartus II 5.0 软件的安装	14
2.2.3 Quartus II 5.0 软件的授权	16
2.3 Quartus II 5.0 软件的设计操作	18
2.3.1 设计输入	19
2.3.2 创建工程	19

2.3.3	建立图形设计文件	19
2.3.4	建立文本编辑文件	32
2.3.5	建立存储器编辑器文件	33
2.4	Quartus II 5.0 设计项目的编译	36
2.4.1	设计综合	37
2.4.2	编译器窗口	37
2.4.3	编译器选项设置	39
2.4.4	引脚分配	45
2.4.5	启动编译器	47
2.4.6	查看适配结果	49
2.5	Quartus II 5.0 设计项目的仿真实验	53
2.5.1	创建一个仿真波形文件	53
2.5.2	设计仿真	56
2.5.3	仿真结果分析	57
2.6	时序分析	58
2.6.1	时序分析基本参数	58
2.6.2	指定时序要求	59
2.6.3	完成时序分析	62
2.6.4	查看时序分析结果	62
2.7	器件编程	63
2.7.1	完成器件编程	64
2.7.2	编程硬件驱动安装	66
第3章	VHDL 语言程序设计	68
3.1	VHDL 概述	68
3.1.1	常用硬件描述语言简介	68
3.1.2	VHDL 及其优点	68
3.1.3	VHDL 程序设计举例	70
3.2	VHDL 程序基本结构	71
3.2.1	实体	72
3.2.2	结构体	74
3.2.3	库	85
3.2.4	程序包	87
3.2.5	配置	91
3.3	VHDL 语言要素	94
3.3.1	VHDL 语言数据对象及其分类	94
3.3.2	VHDL 语言数据类型	98
3.3.3	VHDL 语言运算操作符	104
3.4	VHDL 语言顺序语句	109
3.4.1	赋值语句	110

3.4.2	条件语句	111
3.4.3	case 语句	113
3.4.4	loop 语句	127
3.4.5	next 语句	129
3.4.6	exit 语句	130
3.4.7	wait 语句	132
3.4.8	顺序语句中子程序调用语句	134
3.4.9	返回语句	136
3.4.10	空操作语句	137
3.5	VHDL 并行语句	137
3.5.1	条件信号代入语句	137
3.5.2	选择信号代入语句	139
3.5.3	元件例化语句	141
3.5.4	并行赋值语句(即信号代入语句)	142
3.5.5	生成语句	143
第 4 章	数字系统设计与实现	145
4.1	模为 60 的计数器设计与实现	145
4.1.1	建立图形文件	145
4.1.2	项目编译	146
4.1.3	项目仿真	146
4.2	时钟电路的设计与实现	148
4.2.1	文本编辑法设计模为 24 的计数电路	148
4.2.2	建立顶层 clock 文件与时钟电路设计	149
4.3	状态机电路设计与实现	151
4.3.1	有限状态机的编码规则	151
4.3.2	有限状态机的设计	151
4.3.3	有限状态机的 VHDL 设计	152
4.4	半整数分频器的设计	154
4.4.1	小数分频的基本原理	154
4.4.2	电路组成	154
4.4.3	半整数分频器设计	155
4.5	UART 数据接收发送电路设计与实现	157
4.5.1	波特率的设定	158
4.5.2	数据发送	158
4.5.3	数据接收	159
4.5.4	UART 程序设计	159
4.6	CPLD 在人机接口中的设计与实现	166
4.6.1	接口电路分析与设计	166
4.6.2	接口电路的部分软件设计	168

4.7	并行 8255 接口电路设计与实现	170
4.7.1	并行 8255 接口电路结构分析	171
4.7.2	接口电路 VHDL 实际编程	172
第 5 章	提高电路设计效率的常用方法	202
5.1	引言	202
5.2	EAB 单元的使用	202
5.2.1	EAB 单元结构分析	202
5.2.2	EAB 设计应用	204
5.3	芯片速度的优化	206
5.3.1	修改低层布局	206
5.3.2	合理使用 CPLD 资源	206
5.3.3	通过软件配置提高系统速度	207
5.4	使用 LPM 宏单元库	209
5.5	提高设计效率的综合应用	211
5.5.1	编程过程语法设计效率的提高	211
5.5.2	编程过程中数据类型应用与端口定义	215
第 6 章	大规模可编程逻辑器件	217
6.1	可编程逻辑器件概述	217
6.2	简单可编程逻辑器件	219
6.3	复杂可编程逻辑器件	220
6.3.1	CPLD 的基本结构	221
6.3.2	Altera 公司的器件	221
6.4	现场可编程门阵列	226
6.4.1	FPGA 的整体结构	226
6.4.2	Xilinx 公司的 FPGA 器件	227
6.4.3	FPGA 的配置	230
6.5	在系统可编程逻辑器件	232
6.5.1	ispLSI/pLSI 的结构	232
6.5.2	Lattice 公司的 ispLSI 系列器件	234
6.6	FPGA 和 CPLD 的开发应用选择	235
6.6.1	FPGA 和 CPLD 的性能比较	235
6.6.2	FPGA 和 CPLD 的开发应用选择	236
第 7 章	EDA 技术综合设计应用	239
7.1	数字闹钟的设计	239
7.1.1	系统的设计要求	239
7.1.2	系统的总体设计	239
7.1.3	闹钟控制器的设计	241
7.1.4	译码器的设计	246
7.1.5	键盘缓冲器(预置寄存器)的设计	247

208	7.1.6	闹钟寄存器的设计	248
208	7.1.7	时间计数器的设计	249
209	7.1.8	显示驱动器的设计	251
112	7.1.9	分频器的设计	253
212	7.1.10	系统的整体组装	254
212	7.1.11	系统的硬件验证	255
212	7.2	多功能信号发生器的设计	256
128	7.2.1	设计要求	256
222	7.2.2	设计实现	256
222	7.3	序列检测器的设计	262
227	7.3.1	设计思路	262
228	7.3.2	VHDL 程序实现	262
022	7.3.3	硬件逻辑验证	263
	7.4	交通灯信号控制器的设计	263
	7.4.1	设计思路	263
	7.4.2	VHDL 程序实现	264
	7.4.3	硬件逻辑验证	265
	7.5	电梯控制系统的设计	266
	7.5.1	设计要求	266
	7.5.2	设计实现	266
	7.6	步进电动机控制电路的设计	271
	7.6.1	步进电动机的工作原理	271
	7.6.2	驱动电路的组成及 VHDL 实现	273
	7.7	智力竞赛抢答器的设计	274
	7.7.1	设计思路	274
	7.7.2	VHDL 程序实现	275
	7.8	单片机与 FPGA/CPLD 总线接口的设计	279
	7.8.1	设计思路	280
	7.8.2	VHDL 程序实现	281
第 8 章	EDA 实验开发系统		284
8.1	GW48 型 EDA 实验开发系统原理与使用		284
	8.1.1	系统性能及使用注意事项	284
	8.1.2	系统工作原理	285
	8.1.3	系统主板结构与使用方法	286
8.2	GW48 实验电路结构图		291
	8.2.1	实验电路信号资源符号图说明	291
	8.2.2	各实验电路结构图特点与适用范围简述	292
8.3	GW48 系统结构图信号名与芯片引脚对照表		299
8.4	GW48-B 电子设计竞赛应用板使用说明		303

8.5	GW48 型 EDA 实验开发系统使用实例	305
第 9 章	EDA 技术实验	309
9.1	实验 1 8 位全加器的设计	309
9.2	实验 2 组合逻辑电路的设计	311
9.3	实验 3 触发器功能的模拟实现	313
9.4	实验 4 计数器的设计	315
9.5	实验 5 计数译码显示电路	317
9.6	实验 6 数字钟综合实验	321
9.7	实验 7 序列检测器的设计	322
9.8	实验 8 简易彩灯控制器	323
9.9	实验 9 正负脉宽数控调制信号发生器的设计	325
9.10	实验 10 A/D 转换控制器的设计	327
	参考文献	330

第1章 EDA 技术概述

电子设计自动化(Electronic Design Automation, EDA)技术是以微电子技术发展为物理层面,现代电子设计技术为灵魂,计算机软件技术为手段,最终形成集成电子系统或专用集成电路(Application Specific Integrated Circuit, ASIC)为目的的一门新兴技术。EDA技术的使用对象由两大类人员组成,一类是专用集成电路的芯片设计研发人员;另一类是广大的电子线路设计人员,他们不具备集成电路深层次的知识。本书所阐述的 EDA 技术以后者为应用对象,这样,EDA 技术可简单概括为以大规模可编程逻辑器件为设计载体,通过硬件描述语言输入给相应开发软件,经过编译和仿真,最终下载到设计载体中,从而完成系统电路设计任务的一门新技术。

1.1 EDA 技术及其发展

1.1.1 EDA 技术的涵义

20 世纪末,数字电子技术得到了飞速发展,有力地推动和促进了社会生产力的发展和社会信息化的提高,数字电子技术的应用已经渗透到人类生活的各个方面。从计算机到手机,从数字电话到数字电视,从家用电器到军用设备,从工业自动化到航天技术,都广泛采用了数字电子技术。

微电子技术,即大规模集成电路加工技术的进步是现代数字电子技术发展的基础。目前,在硅片单位面积上集成的晶体管数目越来越多,1978 年推出的 8086 微处理器芯片集成的晶体管数目是 4 万只,到 2000 年推出的 Pentium IV 微处理芯片的集成度达到了 4200 万只晶体管。原来需要成千上万只电子元器件组成的一台计算机主板电路,现在仅用几片超大规模集成电路就可以代替,现代集成电路已经能够实现单片电子系统(System On Chip, SOC)的功能。

现代电子设计技术的核心是 EDA 技术。EDA 技术就是依靠功能强大的电子计算机,在 EDA 工具软件平台上,对以硬件描述语言(Hardware Description Language, HDL)为系统逻辑描述手段完成的设计文件,自动地完成逻辑编译、化简、分割、综合、优化和仿真,直至下载到可编程逻辑器件 CPLD/FPGA 或专用集成电路芯片中,实现既定的电子电路设计功能。EDA 技术使得电路设计者的工作仅限于利用硬件描述语言和 EDA 软件平台来完成对系统硬件功能的实现,极大地提高了设计效率,缩短了设计周期,节省了设计成本。

1.1.2 EDA 技术的发展史

从 20 世纪 60 年代中期开始,人们不断开发出各种计算机辅助设计工具来帮助设计

人员进行集成电路和电子系统的设计,集成电路技术的不断发展对 EDA 技术提出了新的要求,并促进了 EDA 技术的发展。近 30 年来,EDA 技术大致经历了三个发展阶段。

1. CAD 阶段

20 世纪 60 年代中期至 20 世纪 80 年代为计算机辅助设计(Computer-aided Design, CAD)发展的初期。这个阶段人们开始利用计算机取代手工劳动,分别研制了一些单独的软件工具,主要有印制电路板(Printed Circuit Board, PCB)布线设计、电路模拟、逻辑模拟及版图的绘制等,从而可以利用计算机将设计人员从大量繁琐、重复的计算和绘图工作中解脱出来。例如,目前常用的 PCB 布线软件 TANGO 以及用于电路模拟的 SPICE 软件和后来产品化的集成电路版图编辑与设计规划检查系统等软件,都是这个时期的产品。

20 世纪 80 年代初,由于集成电路规模越来越大,制作也越来越复杂,EDA 技术有了较快的发展,许多软件公司,如 Mentor Graphics、Daisy System、Logic System 等进入市场,软件工具的产品开始增多。这个时期的软件主要还是针对产品开发,分为设计、分析、生产、测试等多个独立的软件包。每个软件包只能完成其中的一项工作,但如果通过顺序循环使用这些软件,完成设计的全过程,还存在两方面的问题:首先,由于各个软件工具是由不同的公司和专家开发的,只解决一个领域的问题,若将一个软件工具的输出作为另一个软件工具的输入,就需要人工处理,这往往很繁琐,影响了设计速度;其次,对于复杂电子系统的设计,当时的 EDA 工具不能提供系统级的仿真与综合。由于缺乏系统级的设计考虑,常常在产品开发后期才发现设计有错误,此时再进行修改十分困难。

2. CAED 阶段

20 世纪 80 年代初期至 20 世纪 90 年代初期为计算机辅助工程设计(Computer-aided Engineering Design, CAED)阶段,在集成电路与电子系统设计方法学以及设计工具集成化方面取得了许多成果。各种设计工具,如原理图输入、编译与连接、逻辑模拟、测试码生成、版图自动布局以及各种单元库均已齐全。由于采用了统一数据管理技术,因而能够将各个工具集成为一个 CAED 系统。运用这种系统,按照设计方法学制定的某种设计流程,可以实现由 RT 级开始,从设计输入到版图输出的全程设计自动化。这个阶段中主要采用基于单元库的半定制设计方法,采用门阵列和标准单元法设计的各种 ASIC 得到了极大的发展。多数 CAED 系统中还集成了 PCB 自动布局布线软件以及热特性、噪声、可靠性等分析软件,进而可以实现电子系统设计自动化,这个阶段典型的 CAED 系统有 Mentor Graphics、Valid、Daisy 等公司的产品。

20 世纪 80 年代推出的 EDA 工具以逻辑模拟、定时分析、故障仿真、自动布局和布线为核心,重点解决电路设计完成之前的功能检测等问题。利用这些工具,设计师能在产品制作之前预知产品的功能与性能,能产生产品制造文件,在设计阶段对产品性能的分析就可以进行。

如果说 20 世纪 70 年代的自动布局布线的 CAD 工具代替了设计工作中绘图的重复劳动,那么,20 世纪 80 年代初期的具有自动综合能力的计算机辅助工程(Computer-aided Engineering, CAE)工具则代替了设计师部分工作,对保证电子系统的设计,制造出最佳的电子产品起着关键的作用。到了 20 世纪 80 年代后期,EDA 工具已经可以进行设计描述、综合与优化和设计结果验证。CAED 阶段的 EDA 工具不仅为成功开发电子产品创造了有利条件,而且为高级设计人员的创造性劳动提供了方便。但是,大部分从原理图出发

的 EDA 工具仍然不能适应复杂电子系统的设计要求,并且具体化的元件图形制约着优化设计。

3. EDA 阶段

20 世纪 90 年代以来,微电子技术以惊人的速度发展,其工艺水平已达到深亚微米级,在一个芯片上可以集成数百万乃至上千万晶体管,芯片工作速度可达到 Gb/s,这为制造出规模更大、速度和信息容量更高的芯片系统提供了基础条件,同时也对 EDA 系统提出了更高的要求,并大大促进了 EDA 技术的发展。20 世纪 90 年代以后,出现了以高级语言描述、系统仿真和综合技术为特征的第三代 EDA 技术,它不仅极大地提高了系统的设计效率,而且使设计者摆脱了大量的辅助性工作,将精力集中于创造性的方案与概念的构思上。这个阶段的 EDA 技术主要有如下特征:

(1) 高层综合(High Level Synthesis, HLS)的理论与方法取得进展,从而将 EDA 设计层次由 RT 级提高到系统级(又称为行为级),并且推出了相应的行为级综合优化工具,大大缩短了复杂 ASIC 的设计周期,同时改进了设计质量。典型工具有 Synopsys 公司的 Behavioral Compiler、Mentor Graphics 公司的 Monet 和 Renoir。

(2) 采用硬件描述语言来描述 10 万门以上的设计,并形成了 VHDL 和 Verilog HDL 两种标准硬件描述语言。它们均支持不同层次的描述,使得复杂集成电路的描述规范化,便于传递、交流、保存与修改,并可以建立独立工艺的设计文档,便于设计重用。

(3) 采用平面规划(Floorplaning)技术对逻辑综合和物理版图设计进行联合管理,做到在逻辑综合早期设计阶段就考虑到物理设计信息的影响。通过这些信息,设计者能进行更进一步的综合与优化,并保证所做的修改只会提高性能而不会给版图设计带来负面影响。这对于深亚微米级布线延时已成为主要延时的情况下,加速设计过程的收敛与成功是有所帮助的。在 Synopsys 公司和 Cadence 等公司的 EDA 系统中均采用了这项技术。

(4) 可测性综合设计。随着 ASIC 的规模与复杂性的增加,测试的难度与费用急剧上升,由此产生了将测试电路结构做在 ASIC 芯片上的构想,于是开发了扫描插入、内建自测试(BIST)、边界扫描等可测性设计(DFT)工具,并集成到 EDA 系统中。典型产品有 Compass 公司的 Test Assistant, Mentor Graphics 公司的 LBIST Architect、BSD Architect、DFT Advisor 等。

(5) 为带有嵌入 μP 核的 ASIC 设计提供软、硬件协同设计工具。典型产品有 Mentor Graphics 公司的 Seamless CVE(Co-Verification Environment)等。

(6) 建立并行工程(Concurrent Engineering, CE)框架结构的集成化设计环境,以适应当今 ASIC 的如下特点:规模大而复杂,数字与模拟电路并存,硬件与软件设计并存,产品上市速度要快。该框架可以将不同公司的优秀工具集成为一个完整的 EDA 系统,并能在 Unix 与 Windows NT 两种平台之间平滑过渡。各种 EDA 工具在该框架中可以并行使用。通过统一的集成化设计环境,能够保证各个设计工具之间的相互联系与管理。在这种集成化设计环境中,使用统一的数据管理系统与完善的通信管理系统,由若干个相关的设计小组共享数据库和知识库,同时并行地进行设计。一旦系统设计完成,相应的电路设计、版图设计、可测性设计与嵌入软件的设计等也都基本上完成了。

今天,EDA 技术已经成为电子设计的重要工具,无论是设计芯片还是设计系统,如果没有 EDA 工具的支持,都将是难以完成的。EDA 工具已经成为现代电路设计工程师的

重要武器,正在发挥越来越重要的作用。

1.2 EDA 技术的基本工具

EDA 工具在 EDA 技术应用中占据极其重要的位置,EDA 的核心是利用计算机完成电路设计的全程自动化,因此基于计算机环境下的 EDA 工具软件的支持是必不可少的。EDA 工具的发展经历了两个大的阶段:物理工具和逻辑工具。现在 EDA 和系统设计工具逐步被理解成一个整体的概念:电子系统设计自动化。物理工具用来完成设计中的实际物理问题,如芯片布局、印制电路板布线等;逻辑工具则是基于网表、布尔逻辑、传输时序等概念,首先由原理图编辑器或 HDL 进行设计输入,然后利用 EDA 系统完成综合、仿真、优化等过程,最后生成物理工具可以接受的网表或 VHDL、Verilog HDL 的结构化描述。现在常见的 EDA 工具大致可以分为设计输入编辑器、仿真器、检查/分析工具、优化/综合工具等五个模块。

1.2.1 设计输入编辑器

通常,专业的 EDA 工具供应商或各个可编程逻辑器件厂商提供的 EDA 开发工具,在这些 EDA 开发工具中都含有输入编辑器,如 Xilinx 公司的 Foundation、Altera 公司的 Quartus II、MAX + PLUS II 等。

一般的设计输入编辑器都支持图形输入和 HDL 文本输入。图形输入通常包括原理图输入、状态图输入和波形输入三种常用方式。原理图输入方式沿用传统的数字系统设计方式,即根据设计电路的功能和控制条件,画出设计的原理图或状态图或波形图,然后在设计输入编辑器的支持下,将这些图形输入到计算机中,形成图形文件。图形输入方式形象直观,且不需要掌握硬件描述语言,便于初学或教学演示。但图形输入方式存在没有标准化、图形文件兼容性差及不便于电路模块的移植和再利用等缺点。HDL 文本输入方式与传统的计算机软件语言编辑输入基本一致,在设计输入编辑器的支持下,使用某种硬件描述语言对设计电路进行描述,形成 HDL 源程序。HDL 文本输入方式克服了图形输入方式存在的弊端,为 EDA 技术的应用和发展打开了一个广阔天地。

有的 EDA 设计输入工具把图形设计与 HDL 文本输入相结合,利用 HDL 文本输入通用性的优点和图形输入易学性的优点,实现一个复杂的电路系统的设计。输入编辑器在多样性、易学和通用性方面的功能不断增强,标准 EDA 技术中自动化设计程序不断提高。

1.2.2 仿真器

仿真器有基于元件(逻辑门)的仿真器和 HDL 仿真器,基于元件的仿真器缺乏 HDL 仿真器的灵活性和通用性。在此主要介绍 HDL 仿真器。

在 EDA 技术中,仿真器的地位十分重要,行为模型的表达、电子系统的建模、逻辑电路的验证乃至门级系统的测试,每一步都离不开仿真器的模拟检测。在 EDA 发展的初期,快速地进行电路逻辑仿真是当时的核心问题,即使在现在,各个设计环节的仿真仍然是整个 EDA 工程流程中最重要、最耗时的一个步骤。因此,HDL 仿真器的仿真速度、仿

真的准确性和易用性成为衡量仿真器的重要指标。

按仿真器对设计语言不同的处理方式,可以分为编译型和解释型仿真器。编译型仿真器速度较快,但需要预处理,因此不便即时修改;解释型仿真器的仿真速度一般,可随时修改仿真环境和条件。

按处理的硬件描述语言类型分,HDL 仿真器可分为 VHDL 仿真器、Verilog 仿真器、Mixed HDL 仿真器(混合 HDL 仿真器,同时处理 VHDL 与 Verilog HDL)、其他 HDL 仿真器(针对其他 HDL 的仿真)。

按仿真时是否考虑硬件延时分类,HDL 仿真器可分为功能仿真器和时序仿真器。根据输入和仿真文件的不同,可以由不同的仿真器完成,也可由同一仿真器完成。

几乎各个 EDA 厂商都提供基于 VHDL/Verilog HDL 的仿真器。常用的 HDL 仿真器有 Model Technology 公司的 ModelSim、Cadence 公司的 Verilog-XL 和 NC-Sim、Aldec 公司的 Active HDL、Synopsys 公司的 VCS 等。

1.2.3 HDL 综合器

由于目前通用的 HDL 语言有 VHDL 和 Verilog HDL,这里介绍的 HDL 综合器主要是针对这两种语言。

HDL 诞生的初衷是为了电路逻辑的建模和仿真,但直到 Synopsys 公司推出了 HDL 综合器之后,HDL 才被直接用于电路的设计。

HDL 综合器是一种将硬件描述语言转化为硬件电路的重要工具软件。在用 EDA 技术进行电路设计时,HDL 综合器完成电路化简、算法优化和硬件结构细化等操作。HDL 综合器把可综合的 VHDL/Verilog HDL 转化为硬件电路时,一般要经过两个步骤:

第一步是 HDL 综合器对 VHDL/Verilog HDL 进行分析处理,并将其转化成相应的电路结构或模块,这时是不考虑实际器件实现的,即完全与硬件无关,这个过程是一个通用电路原理图形成的过程。

第二步是对应实际实现的目标器件的结构进行优化,并使之满足各种约束条件,优化关键路径等。

HDL 综合器的输出文件一般是网表文件,如 EDIF 格式,文件后缀 .edf 是一种用于设计数据交换和交流的工业标准化格式的文件,或是直接用 VHDL/Verilog HDL 表达的标准格式的网表文件,或是对应现场可编程门阵列(Field Programmable Gate Array,FPGA)器件厂商的网表文件,如 Xilinx 公司的 XNF 网表文件。

由于 HDL 综合器只完成 EDA 设计流程中的一个独立设计步骤,所以它往往被其他 EDA 环境调用,以完成全部流程。EDA 综合器的调用具有前台模式和后台模式两种,用前台模式调用时,可以从计算机的显示器上看到调用窗口界面;用后台模式(也称为控制模式)调用时,不出现图形窗口界面,仅在后台运行。

HDL 综合器的使用也有两种模式:图形模式和命令模式。

1.2.4 适配器(布局、布线器)

适配也称为结构综合,适配器的任务是完成在目标系统器件上的布局、布线。适配通常都由可编程器件厂商提供的专用软件来完成。这些软件可以单独运行或嵌入到厂商提

供的适配器中,但同时提供性能良好、使用方便的专用适配器运行环境,如 IspEXPERT-Compiler。而 Altera 公司的 EDA 集成开发环境 Quartus II、Quartus 中都含有嵌入的适配器,Xilinx 公司的 Foundation 和 IsE 中也同样含有自己的适配器。

适配器最后输出的是各厂商自己定义的下栽文件,用于下载到器件中以实现电路设计。

1.2.5 下载器(编程器)

下载器的任务是把电路设计结果下载到实际器件中,实现硬件设计。下载软件一般由可编程逻辑器件厂商提供,或嵌入到 EDA 开发平台中。

1.3 EDA 技术的基本设计思路

1.3.1 EDA 电路级设计

设计人员首先确定设计方案,并选择能实现该方案的合适元器件,然后根据元器件设计电路原理图,进行第一次仿真,其中包括数字电路的逻辑模拟、故障分析等。其作用是在元件模型库的支持下检验设计方案在功能方面的正确性。

仿真通过之后,根据原理图产生电器连接网表,进行 PCB 的自动布局布线。在制作 PCB 之前,还可以进行 PCB 分析,并将分析结果反馈回电路图,进行第二次仿真,称之为后仿真。其作用是检验 PCB 在实际工作环境中的可行性。

综上所述,EDA 技术的电路级设计可以使设计人员在实际的电子系统产生之前,就“已经”全面了解系统的功能特性和物理特性,从而将开发风险消灭在设计阶段。缩短开发周期,降低开发成本。

1.3.2 EDA 系统级设计

随着技术的进步,电子产品更新换代频繁,产品的复杂程度大幅度增加,以前的电路级设计的 EDA 技术已经不能适应新形势,必须有一种高层次的设计,即“系统级设计”来代替,其设计流程图如图 1.1 所示。

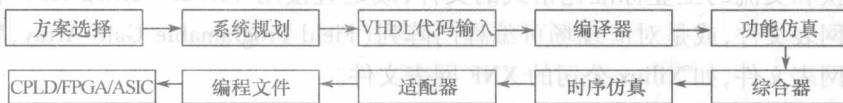


图 1.1 EDA 系统级设计流程图

基于系统级的 EDA 设计方法其主要思路是采用自顶向下的设计方法,使开发者从一开始就要考虑到产品生产周期的诸多方面,包括质量成本、开发周期等因素。第一步从系统方案设计入手,在顶层进行系统功能划分和结构设计;第二步用 VHDL、Verilog HDL 等硬件描述语言对高层次的系统行为进行描述;第三步通过编译器形成标准的 VHDL 文件,并在系统级验证系统功能的设计正确性;第四步用逻辑综合优化工具生成具体的门级电路的网表,这是将高层次描述转化为硬件电路的关键;第五步利用产品的网表进行适配前的时序仿真;最后是系统的物理体现,它可以是复杂可编程逻辑器件(Complex Program-

able Logic Device, CPLD)、FPGA 或 ASIC。

1.4 EDA 的设计流程

利用 EDA 技术进行电路设计的大部分工作是在 EDA 软件工作平台上进行的,EDA 设计流程图如图 1.2 所示。EDA 设计流程包括设计准备、设计输入、设计处理、器件编译和设计完成五个步骤,以及相应的功能仿真、时序仿真和器件测试三个设计验证过程。

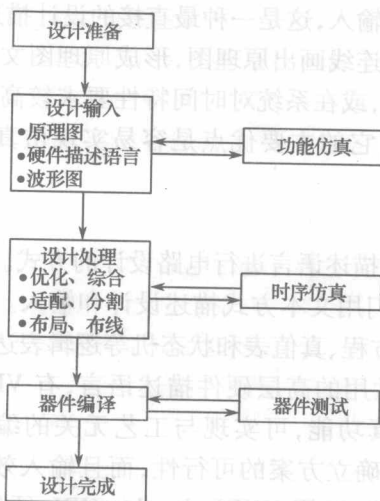


图 1.2 EDA 设计流程

1.4.1 设计准备

在对可编程逻辑器件的芯片进行设计之前,首先要进行方案论证、系统设计和器件选择等设计准备工作。设计者首先要根据任务要求,如系统所完成的功能及复杂程度,对工作速度和器件本身的资源、成本及连线的可行性等方面进行权衡,选择合适的设计方案和合适的器件类型。

数字系统的设计有多种方法,如模块设计法、自顶向下设计法和自底向上设计法等。自顶向下设计法是目前最常用的设计方法,也是基于芯片的系统设计的主要方法。它首先从系统设计入手,在顶层进行功能划分和结构设计,采用 HDL 对高层次的系统进行描述,并在系统级采用仿真手段验证设计的正确性,然后再逐级设计低层的结构。由于高层次的设计与器件及工艺无关,而且在芯片设计前就可以用软件仿真手段验证系统方案的可行性,因此自顶向下的设计方法有利于在早期发现结构设计中的错误,避免不必要的重复设计,提高设计的一次成功率。

自顶向下的设计采用功能分割的方法从顶向下逐次进行划分。在设计过程中采用层次化和模块化将使系统设计变得简洁和方便。层次化设计时分层次、分模块地进行设计描述。描述器件总功能的模块放在最上层,称之为顶层设计;描述器件某一部分功能的模块放在下层,称之为底层设计;底层模块还可以向下再分层,这种分层关系类似于软件设计中的主程序和子程序的关系。层次化设计的优点一是支持模块化,底层模块可以反复