



普通高等教育电气电子类工程应用型“十二五”规划教材

EDA 技术及应用

孙宏国 周磊 编著 ·····



机械工业出版社
CHINA MACHINE PRESS

普通高等教育电气电子类工程应用型“十二五”规划教材

EDA 技术及应用

孙宏国 周磊 编 著



机械工业出版社

本书共6章,第1章讲述了EDA技术的特点、概念和数字系统的设计方法;第2章介绍了VHDL语言的基本语法特点、程序结构、常用语句以及相关基础知识;第3章介绍了一些典型的基本门电路、组合逻辑电路和时序逻辑电路VHDL语言的实现方式;第4章介绍了Altera公司的综合开发软件的特点和使用方法;第5章介绍了EDA技术在不同专业的工程应用案例;第6章列举了20个基础性和综合性实验项目;附录A介绍了YCIT_SOPC_V1.0实验开发板;附录B介绍了可编程逻辑器件及其配置。

本书可作为高等学校电气类、电子信息类、自动化类、计算机类本、专科专业的“EDA技术”课程的教材,亦可作为参加电子设计竞赛的培训教材和参考书,还可作为电子爱好者的自学教程。

本书免费配有电子课件,欢迎选用本书作为教材的老师登录www.cmpedu.com注册下载。

图书在版编目(CIP)数据

EDA技术及应用/孙宏国,周磊编著. —北京:机械工业出版社,2013.1

普通高等教育电气电子类工程应用型“十二五”规划教材

ISBN 978-7-111-40434-7

I. ①E… II. ①孙…②周… III. ①电子电路-电路设计-计算机辅助设计-高等学校-教材 IV. ①TN702

中国版本图书馆CIP数据核字(2012)第273922号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑:贡克勤 责任编辑:贡克勤 王保家

版式设计:霍永明 责任校对:李锦莉

封面设计:张静 责任印制:张楠

北京京丰印刷厂印刷

2013年1月第1版·第1次印刷

184mm×260mm·22.75印张·1插页·562千字

标准书号:ISBN 978-7-111-40434-7

定价:45.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

社服务中心:(010) 88361066

销售一部:(010) 68326294

销售二部:(010) 88379649

读者购书热线:(010) 88379203

网络服务

教材网:<http://www.cmpedu.com>

机工官网:<http://www.cmpbook.com>

机工官博:<http://weibo.com/cmp1952>

封面无防伪标均为盗版

目 录

前言

第 1 章 EDA 技术概述 1

- 1.1 EDA 技术的涵义 1
- 1.2 EDA 技术的发展历程 1
- 1.3 EDA 技术的主要内容 3
- 1.4 EDA 软件系统的构成 5
- 1.5 EDA 的工程设计流程 6
- 1.6 数字系统的设计 10
 - 1.6.1 数字系统的设计模型 10
 - 1.6.2 数字系统的设计方法 11
 - 1.6.3 数字系统的设计准则 11
 - 1.6.4 数字系统的设计步骤 12

习题 13

第 2 章 VHDL 语言程序基础 14

- 2.1 概述 14
- 2.2 VHDL 语言程序的结构 15
 - 2.2.1 VHDL 语言程序设计的基本单元 15
 - 2.2.2 实体 16
 - 2.2.3 构造体 18
 - 2.2.4 配置 19
- 2.3 VHDL 设计资源 21
 - 2.3.1 库 21
 - 2.3.2 包集合 23
- 2.4 VHDL 语言要素 24
 - 2.4.1 标志符 24
 - 2.4.2 数据对象 25
 - 2.4.3 VHDL 的数据类型 29
 - 2.4.4 VHDL 运算符 36
 - 2.4.5 VHDL 的属性 38
 - 2.4.6 常见错误 39
- 2.5 VHDL 语言的描述方式 40
 - 2.5.1 行为描述 41
 - 2.5.2 数据流描述 42
 - 2.5.3 结构描述 42
- 2.6 VHDL 顺序语句 43
 - 2.6.1 赋值语句 44

2.6.2 转向控制语句 45

2.6.3 等待语句 51

2.6.4 子程序调用语句 52

2.6.5 返回语句 53

2.6.6 空操作语句 53

2.6.7 其他语句 54

2.7 VHDL 并行语句 56

2.7.1 进程语句 56

2.7.2 块语句 59

2.7.3 并行信号赋值语句 61

2.7.4 并行过程调用语句 63

2.7.5 元件例化语句 63

2.7.6 生成语句 67

2.8 子程序 70

2.8.1 函数 70

2.8.2 重载函数 70

2.8.3 过程 70

2.8.4 重载过程 70

习题 70

第 3 章 基本逻辑单元的 VHDL

模型 7

3.1 组合逻辑电路设计 7

3.1.1 基本逻辑门设计 7

3.1.2 编码器、译码器和数据选择器 7

3.1.3 加法器 8

3.1.4 三态门及总线缓冲器 8

3.1.5 运算电路 8

3.2 时序逻辑电路设计 9

3.2.1 触发器 9

3.2.2 寄存器 9

3.2.3 计数器 9

3.2.4 分频器 10

3.2.5 序列信号发生器和检测器 10

3.3 存储器 10

3.3.1 存储器描述中的一些共性问题 10

3.3.2 只读存储器 10

3.3.3 随机存储器	111	应用	284
3.3.4 堆栈	112	5.4.1 系统框图	285
3.4 有限状态机	117	5.4.2 功率因数和低压系统工作频率 的测量	285
3.4.1 有限状态机的分类	118	5.4.3 晶闸管触发脉冲产生电路	290
3.4.2 有限状态机的应用	122	5.4.4 误差分析	293
习题	125	5.4.5 小结	293
第4章 Quartus II 介绍与使用	127	5.5 数字钟的 EDA 设计	294
4.1 概述	127	5.5.1 设计要求	294
4.2 创建第一个工程	127	5.5.2 功能描述	294
4.2.1 使用工程向导创建工程	127	5.5.3 各单元电路的实现	294
4.2.2 完善顶层设计文件	131	5.5.4 数字钟的顶层电路设计	300
4.2.3 Quartus II 的交叉探索	145	第6章 EDA 技术实验	303
4.2.4 约束设计	150	6.1 Quartus II 的使用	303
4.2.5 输入/输出端口配置	151	6.2 7 人表决器	304
4.3 设计的时序约束	157	6.3 格雷码变换电路	305
4.4 团队协作及逻辑锁定	161	6.4 BCD 码加法器	306
4.5 ModelSim 仿真	171	6.5 4 位全加器	308
4.6 SignalTap II 的使用	180	6.6 英语字母显示电路	309
4.7 时序逼近	192	6.7 4 位并行乘法器	310
第5章 EDA 技术工程应用实例	217	6.8 设计基本触发器	311
5.1 SOPC 设计	217	6.9 设计 74LS160 计数器功能模块	312
5.1.1 概述	217	6.10 步长可变的加减计数器	313
5.1.2 Hello World 工程	222	6.11 可控脉冲发生器	314
5.1.3 EPCS + FPGA + SDRAM 结构	244	6.12 正负脉宽数控调制信号发生器	315
5.1.4 小结	246	6.13 序列检测器	316
5.2 基于 FPGA 的 PID 算法实现	247	6.14 4 位移位乘法器	316
5.2.1 概述	247	6.15 出租车计费器	318
5.2.2 数字 PID 控制器	248	6.16 数字秒表	320
5.2.3 PID 算法的程序流程	248	6.17 频率计	321
5.2.4 FPGA 实现 PID 的原理	249	6.18 交通灯控制器	322
5.2.5 构建 PID 模块	251	6.19 数码锁	324
5.2.6 软 PID 控制的实现	264	6.20 乒乓球游戏机	326
5.3 8031 内核设计	274	附录	329
5.3.1 ALU 部分	277	附录 A YCIT_SOPC_V1.0 实验开发板 简介	329
5.3.2 定时器/计数器	280	附录 B 可编程逻辑器件简介及其配置	330
5.3.3 中断	282	参考文献	35
5.3.4 串行接口	282		
5.3.5 RAM 和 ROM	284		
5.4 EDA 在无功补偿控制系统中的			

第 1 章 EDA 技术概述

1.1 EDA 技术的涵义

什么叫 EDA 技术？它是一门迅速发展的高新技术，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，自动完成用软件方式设计的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。

利用 EDA 技术进行电子系统的设计，具有以下几个特点：①用软件的方式设计硬件；②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；③设计过程中可用有关软件进行各种仿真；④系统可现场编程，在线升级；⑤整个系统可集成在一个芯片上，体积小、功耗低、可靠性高。因此，EDA 技术是现代电子系统设计的发展趋势并将成为主流。

1.2 EDA 技术的发展历程

EDA 技术伴随着计算机、集成电路、电子系统设计的发展，经历了计算机辅助设计（Computer Aided Design, CAD）、计算机辅助工程设计（Computer Aided Engineering Design, CAE）和电子设计自动化（Electronic Design Automation, EDA）三个发展阶段。

1. 20 世纪 70 年代的计算机辅助设计（CAD）阶段

早期的电子系统硬件设计采用的是分立元件，随着集成电路的出现和应用，硬件设计进入到发展的初级阶段。初级阶段的硬件设计大量选用中小规模标准集成电路，人们将这些器件焊接在电路板上，做成初级电子系统，对电子系统的调试是在组装好的印制电路板（Printed Circuit Board, PCB）上进行的。

由于设计师对图形符号使用数量有限，传统的手工布图方法无法满足产品复杂性的要求，更不能满足工作效率的要求。这时，人们开始将产品设计过程中高度重复性的繁杂劳动（如布图布线工作），用二维图形编辑与分析的 CAD 工具替代，最具代表性的产品就是美国 ACCEL 公司开发的 Tango 布线软件。20 世纪 70 年代，是 EDA 技术发展初期，由于 PCB 布图布线工具受到计算机工作平台的制约，其支持的设计工作有限且性能比较差。

2. 20 世纪 80 年代的计算机辅助工程设计（CAE）阶段

初级阶段的硬件设计是用大量不同型号的标准芯片实现电子系统设计的。随着微电子工艺的发展，相继出现了集成上万只晶体管的微处理器、集成几十万直到上百万储存单元的随机存储器（Random Access Memory, RAM）和只读存储器（Read-Only Memory, ROM）。此

外,支持定制单元电路设计的硅编辑、掩模编程的门阵列,如标准单元的半定制设计方法以及可编程逻辑器件(Programmable Array Logic, PAL 和 Generic Array Logic, GAL)等一系列微结构和微电子学的研究成果都为电子系统的设计提供了新天地。因此,可以用少数几种通用的标准芯片实现电子系统的设计。

伴随计算机和集成电路的发展,EDA 技术进入到计算机辅助工程设计阶段。20 世纪 80 年代初,推出的 EDA 工具则以逻辑模拟、定时分析、故障仿真、自动布局和布线为核心,重点解决电路设计没有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,能生成产品制造文件,在设计阶段对产品性能的分析前进了一大步。

如果说 20 世纪 70 年代的自动布局布线的加工具代替了设计工作中绘图的重复劳动,那么,到了 80 年代出现的具有自动综合能力的 CAE 工具则代替了设计师的部分工作,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 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,CAE 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,而具体化的元件图形制约着优化设计。

3. 20 世纪 90 年代电子系统设计自动化(EDA)阶段

为了满足千差万别的系统用户提出的设计要求,最好的办法是由用户自己设计芯片,让他们把想设计的电路直接设计在自己的专用芯片上。微电子技术的发展,特别是可编程逻辑器件(Programmable Logic Device, PLD)的发展,使得微电子厂家可以为用户提供各种规模的可编程逻辑器件,使设计者通过设计芯片实现电子系统功能。EDA 工具的发展,又为设计师提供了全线 EDA 工具。这个阶段发展起来的 EDA 工具,目的是在设计前期将设计师从事的许多高层次设计由工具来完成,如可以将用户要求转换为设计技术规范,有效处理可用设计资源与理想设计目标之间的矛盾,按具体的硬件、软件和算法进行设计分解等。由于电子技术和 EDA 工具的发展,设计师可以在不太长的时间内使用 EDA 工具,通过一些简单标准化的设计过程,利用微电子厂家提供的设计库来完成数万门专用集成电路(Application Specific Integrated Circuit, ASIC)和集成系统的设计与验证。

20 世纪 90 年代,设计师逐步从使用硬件转向设计硬件,从单个电子个品开发转向系统级电子产品开发,即片上系统集成(System on Chip, SOC)。因此,EDA 工具是以系统级设计为核心,包括系统行为级描述与结构综合、系统仿真与测试验证、系统划分与指标分配、系统决策与文件生成等一整套的电子系统设计自动化工具。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与厂家的系统级设计能力,具有高级抽象的设计构思手段。例如,提供框图、状态图和流程图的编辑能力,具有适合层次描述和混合信号描述的硬件描述语言(Hardware Description Language, HDL),同时含有各种工艺的标准元件库。只有具备上述功能的 EDA 工具,才可能使电子工程师在不熟悉各种半导体工艺的情况下,完成电子系统的设计。

未来 EDA 技术将向广度和深度两个方向发展,EDA 将会超越电子设计的范畴进入其他领域,软、硬核(Core)功能库的建立,以及基于自顶向下的电子系统设计理念的确定,EDA 技术将是 21 世纪产生重大影响的十大技术之一。

1.3 EDA 技术的主要内容

EDA 技术涉及面广，内容丰富，从教学和实用的角度看，究竟应掌握些什么内容呢？主要应掌握如下 4 个方面的内容：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自动化的设计工具，实验开发系统则是利用 EDA 技术进行电子系统设计的下载工具及软件验证工具。为了便于读者对 EDA 技术有一个总体印象，下面对 EDA 技术的主要内容进行概要的介绍。

1. 大规模可编程逻辑器件

可编程逻辑器件 (PLD) 是一种由用户编程以实现某种逻辑功能的新型逻辑器件。FPGA 和 CPLD 分别是现场可编程门阵列和复杂可编程逻辑器件的简称，现在，FPGA 和 CPLD 的应用已十分广泛，它们将随着 EDA 技术的发展而成为电子设计领域的重要角色。国际上生产 FPGA/CPLD 的主流公司，并且在国内占有市场份额较大的主要是 Xilinx、Altera、Lattice 三家公司。Xilinx 以 CoolRunner、XC9500 系列为代表的 CPLD，以及以 XC4000、Spartan、Virtex 系列为代表的 FPGA 器件，如 C2000、XC4000、Spartan 和 Virtex、Virtex II pro、Virtex-4、Virtex-6、Spartan-6 等系列，其性能不断提高。Altera 公司提供的主要可编程逻辑器件系列有 Classic 系列、MAX (Multiple Array Matrix) 系列、FLEX (Flexible Logic Element Matrix) 系列、APEX (Advanced Logic Array Matrix) 系列、ACEX 系列、APEX II 系列、Cyclone 系列、Stratix 系列、MAX II 系列、Cyclone II 系列以及 Stratix II 系列等。Lattice 公司的 ISP-PLD 器件有 isPLSI1000、isPLSI2000、isPLSI3000、isPLSI6000 系列等。

FPGA 在结构上主要分为三个部分，即可编程逻辑单元、可编程输入/输出单元和可编程连线。CPLD 在结构上主要包括三个部分，即可编程逻辑宏单元、可编程输入/输出单元和可编程内部连线。

高集成度、高速度和高可靠性是 FPGA/CPLD 最明显的特点，其时钟延时可小至 ns 级，结合其并行工作方式，在超高速应用领域和实时测控方面有着非常广阔的应用前景。在高可靠应用领域，如果设计得当，将不会存在类似于 MCU 的复位不可靠和 PC 可能跑飞等问题。FPGA/CPLD 的高可靠性还表现在几乎可将整个系统下载于同一芯片中，实现所谓片上系统，从而大大缩小了体积，易于管理和屏蔽。

由于 FPGA/CPLD 的集成规模非常大，故可利用先进的 EDA 工具进行电子系统设计和产品开发。由于开发工具的通用性、设计语言的标准化以及设计过程几乎与所用器件的硬件结构无关，因而设计开发成功的各类逻辑功能块软件有很好的兼容性和可移植性。它几乎可用于任何型号和规模的 FPGA/CPLD 中，从而使得产品设计效率大幅度提高。可以在很短时间内完成十分复杂的系统设计，这正是产品快速进入市场最宝贵的特征。美国 IT 公司认为，一个 ASIC 的 80% 功能可用于 IP 核 (Intellectual Property core) 等现成逻辑合成。而未来大系统的 FPGA/CPLD 设计仅仅是各类再应用逻辑与 IP 核的拼装，其设计周期将更短。

与 ASIC 设计相比，FPGA/CPLD 显著的优势是开发周期短、投资风险小、产品上市速度快、市场适应能力强和硬件升级回旋余地大，而且当产品定型和产量扩大后，可将在生产

中达到充分检验的 VHDL 设计迅速实现 ASIC 投产。

对于一个开发项目，究竟是选择 FPGA 还是选择 CPLD 呢？主要看开发项目本身的需要。对于普通规模，且产量不是很大的产品项目，通常使用 CPLD 比较好。对于大规模的逻辑设计 ASIC 设计，或单片系统设计，则多采用 FPGA。另外，FPGA 掉电后将丢失原有的逻辑信息，所以在实用中需要为 FPGA 芯片配置一个专用 ROM。

2. 硬件描述语言

使用硬件描述语言，在 EDA 软件提供的设计向导或语言助手的支持之下进行设计，是目前工程设计最主要的设计方法。近年来广泛使用 HDL 的有 ABEL、AHDL、VHDL 和 Verilog HDL。VHDL 和 Verilog HDL 是两种最常用的硬件描述语言。

(1) VHDL VHDL (Very High Speed Integrated Circuit HDL) 即超高速集成电路硬件描述语言，是随着集成电路系统化和高集成化发展起来的，是一种用于数字电子系统的设计和测试方法的描述语言。它是由美国国防部发起、开发并标准化，1987 年公布为 IEEE 标准 (IEEE STD1076—1987 [LRM87])，1993 年 VHDL 重新修订，形成新的标准，即 IEEE STD1076—1993 [LRM93]。1996 年，IEEE 1076.3 成为 VHDL 的综合标准。

VHDL 语言设计技术齐全、方法灵活、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。该语言较早被引入我国，已经被我国许多高校所接受。1995 年我国国家技术监督局制定《CAD 通用技术规范》推荐 VHDL 作为我国电子设计自动化硬件描述语言的国家标准。掌握 VHDL，利用 VHDL 设计电子电路，是当前进行技术竞争的一项技能和强有力的工具。

VHDL 是语法非常严格的语言，同时，对于同一功能的模块，描述方法也可以有各种形式，因此，VHDL 对于初学者有一定的难度，但对高级用户来说，却是强有力的编程语言。

(2) Verilog HDL Verilog HDL 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，它是由美国 GDA (Gateway Design Automation) 公司的 Philmoorby 在 1983 年末首创的，最初只设计了一个仿真与验证工具，之后以陆续开发了相关的故障模拟与时序分析工具。1989 年 Cadence 公司收购 GDA 公司，并于 1990 年公开发表了 Verilog HDL，成立了 OVI (Open Verilog International) 组织来负责该语言的发展，由于该语言的优越性，各大半导体器件公司纷纷采用它作为开发本公司产品的工具，IEEE 也于 1995 年将它定为协会的标准 (即 IEEE 1364—1995)，且现正在制定关于模拟电路的 Verilog HDL 标准。

Verilog HDL 虽然是硬件描述语言，但其风格与 C 语言非常相近，对已具有 C 语言编程基础的读者，掌握这种语言是很容易的。它适用于 RTL 级和门级的描述，其综合过程较 VHDL 稍简单，较自由的语法也容易使初学者犯一些错误，但其在高级描述方面不如 VHDL。与之相比，VHDL 语言的学习要困难一些。

3. 软件开发工具

目前比较流行的、主流厂家的 EDA 的软件工具有 Altera 公司的 Maxplus II、Quartus II，Lattice 公司的 ispEXPERT，Xilinx 公司 Foundation Series。

这几种软件的基本功能相同，主要差别在于：面向的目标器件不一样 (各公司有自己的系列产品)。它们的性能各有优劣。

4. 实验开发系统

实验开发系统提供芯片下载电路及 EDA 实验/开发的外围资源，以供硬件验证用。主要

包括：①实验或开发所需的各类基本信号发生模块，包括时钟、脉冲、高低电平等等；②FPGA/CPLD 输出信息显示模块，包括数码显示、发光管显示、声响显示或液晶（LCD）显示；③监控程序模块，提供“电路重构软配置”；④目标芯片适配座以及上面的 FPGA/CPLD 目标芯片和编程下载电路。

国内有许多高校和公司开发并向学校推广的实验开发系统，这些开发系统大体相仿，功能基本大同小异，能够满足一般实验教学的要求，但在使用和系统或软件升级方面，尤其是在实际应用方面不是很方便，给学生的印象就是一个类似于其他实验的实验箱，对箱内电路没有任何概念，对可编程逻辑器件的应用不甚清楚，或者不能完全适用不同层次学校的要求，因此我们自行开发了 YCIT_SOPC_V1.0 实验系统，具体介绍见附录 A。

1.4 EDA 软件系统的构成

EDA 技术研究的对象是电子设计的全过程，有系统级、电路级和物理级 3 个层次的设计。其涉及的电子系统从低频、高频到微波，从线性到非线性，从模拟到数字，从通用集成电路到专用集成电路，因此 EDA 技术研究的范畴相当广泛。如果从专用集成电路（ASIC）开发与应用角度看，EDA 软件系统应当包含以下子模块：设计输入子模块、设计数据库子模块、分析验证子模块、综合仿真子模块、布局布线子模块等。

(1) 设计输入子模块 该模块接受用户的设计描述，并进行语义正确性、语法规则的检查。检查通过后，将用户的设计描述数据转换为 EDA 软件系统的内部数据格式，存入设计数据库被其他子模块调用。设计输入子模块不仅能接受图形描述输入、硬件描述语言（HDL）描述输入，还能接受图文混合描述输入。该子模块一般包含针对不同描述方式的编辑器，如图形编辑器、文本编辑器等，同时包含对应的分析器。

(2) 设计数据库子模块 该模块存放系统提供的库单元以及用户的设计描述和中间设计结果。

(3) 分析验证子模块 该模块包括各个层次的模拟验证、设计规则的检查、故障诊断等。

(4) 综合仿真子模块 该模块包括各个层次的综合工具，理想的情况是从高层次到低层次的综合仿真全部由 EDA 工具自动实现。

(5) 布局布线子模块 该模块实现由逻辑设计到物理实现的映射，因此与物理实现的方式密切相关。例如，最终的物理实现可以是门阵列、可编程逻辑器件等，由于对应的器件不同，因此各自的布局布线工具会有很大的差异。

近些年，许多生产可编程逻辑器件的公司都相继推出适于开发自己公司器件的 EDA 工具，这些工具一般都具有上面提到的各个模块，操作简单，对硬件环境要求低，运行平台是 PC 和 Windows 操作系统。如 Xilinx、Altera、Lattice、Actel、AMD 等器件公司都有自己的 EDA 工具。

有的 EDA 软件是由专业 EDA 软件商提供的，称第三方设计软件，例如：目前比较著名的 EDA 综合器有 Synopsys 公司的 FPGA Compiler、FPGA Express；Synplicity 公司的 Synplify；Mentor Graphics 公司的 Autologic II；Data I/O 公司的 Synario。目前，器件生产厂家往往委托专业 EDA 软件商开发或共同开发设计输入、模拟验证和编程等软件，器件生产厂家只研制

适合自身器件要求的编译或转换程序，所以第三方设计软件往往能够开发多家公司的器件，但在设计具体型号的器件时，需要器件制造商提供器件库和适配器（Fitter）软件。表 1-1 给出了目前应用较为广泛的几种 EDA 软件。

表 1-1 几种应用较为广泛的 EDA 软件

公司名称	软件名称	网 址
Altera	Maxplus II/Quartus II	http://www.altera.com
Xilinx	Foundation	http://www.xilinx.com
Data I/O	Synario	http://www.dataio.com
Mentor Graphics	Autologic II	http://www.mentor.com
Cadence Design	FPGA Station	http://www.cadence.com
Synopsys	FPGA Express	http://www.Synopsys.com
Viewlogic	Powerview Tools	http://www.viewlogic.com
Lattice	Isp Expert System/Synario	http://www.latticeemi.com

如何选用这些工具，对于电子系统设计师是十分重要的。一般而言，各类 EDA 软件各有其特点和使用范围，不能一概而论。但是，作为一个优秀的 EDA 设计软件至少应具备如下品质：

- 1) 良好的人机界面，便于使用。
- 2) 集成多种设计方法，尤其重要的是原理图设计、语言设计、并易于与其他 EDA 软件容易交换数据。
- 3) 提供较为充分的元件库和模块，且元件库容易扩充。
- 4) 集成项目管理和各种设计编辑工具，设计、仿真、优化各项功能无缝连接。
- 5) 快速编译和重新编译一种设计。
- 6) 使使用者不受内部器件体系结构细节的影响。
- 7) 几乎不需要人工干预而获得很好的性能。
- 8) 能获得网上在线支持。

1.5 EDA 的工程设计流程

EDA 工程设计流程图如图 1-1 所示，现具体说明如下。

1. 源程序的编辑和编译

利用 EDA 技术进行一项工程设计，首先需利用 EDA 工具的文本编辑器或图形编辑器将它用文本方式或图形方式表达出来，进行排错编译，变成 VHDL 文件格式，为进一步的逻辑综合做准备。

常用的源程序输入方式有以下 4 种：

(1) 原理图设计 原理图设计是 EDA 工具软件提供的基本设计方法。该方法是选用 EDA 软件提供的器件库资源，并利用电路作图的方法，进行相关的电气连接而构成相应的系统或满足某些特定功能的系统或新元件。这种方式大多用在对系统及各部分电路很熟悉的

情况，或在系统对时间特性要求较高的场合。它的主要优点是容易实现仿真，便于信号的观察和电路的调整。原理图设计方法直观、易学，但当系统功能较复杂时，原理图输入方式效率低，它适应不太复杂的小系统和复杂系统的综合设计（与其他设计方法进行联合设计）。原理图设计的编辑窗口示意图如图 1-2 所示。

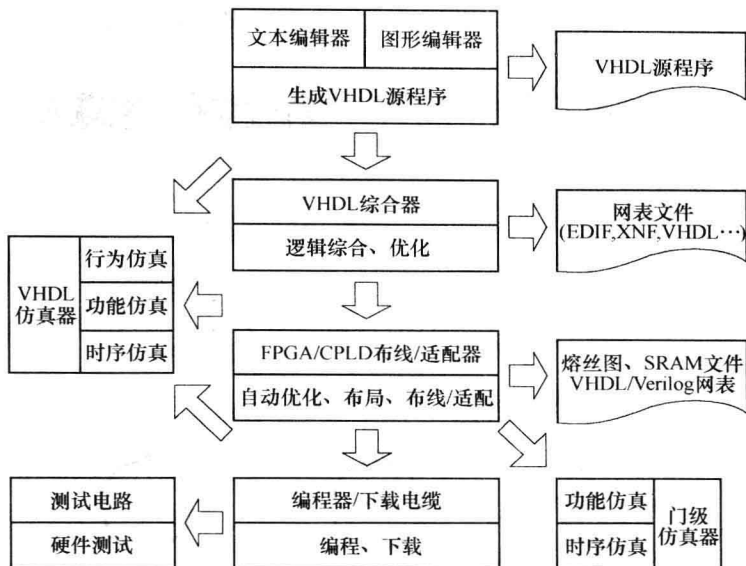


图 1-1 EDA 工程设计流程图

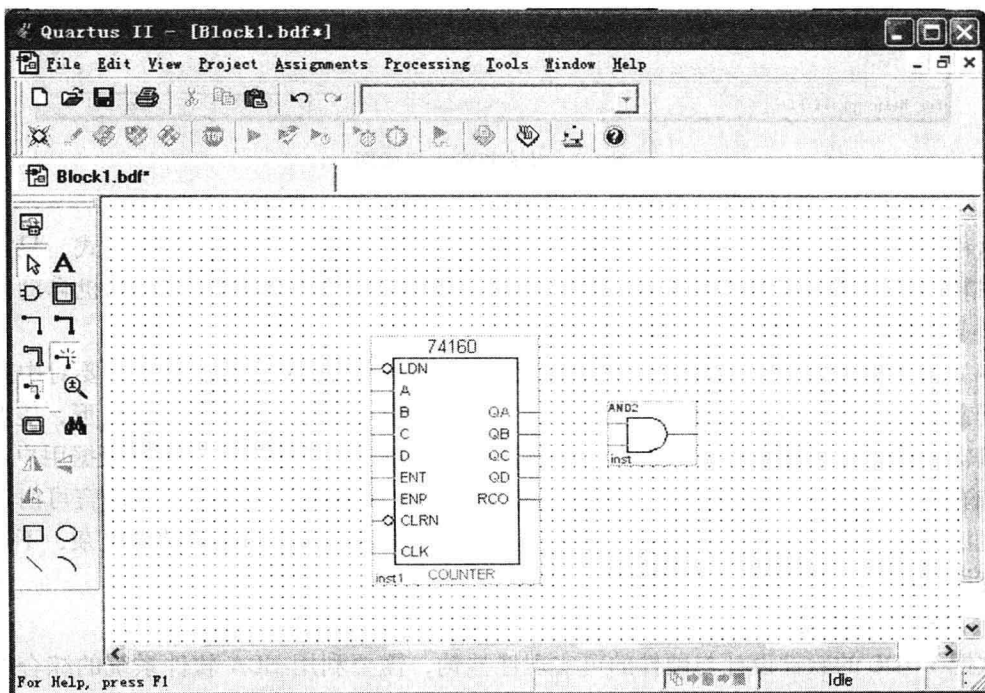


图 1-2 原理图设计的编辑窗口示意图

(2) 程序设计 程序设计是使用硬件描述语言, 在 EDA 软件提供的设计向导或语言助手的支持之下进行设计。HDL 设计是目前工程设计最主要的设计方法。程序设计的语言种类较多, 近年来广泛使用的有 ABEL、AHDL、VHDL 和 Verilog HDL。VHDL 和 Verilog HDL 是两种最常用的硬件描述语言。

(3) 状态机设计 一些 EDA 软件提供了可视化图形状态机输入法, 可以像绘画似地创建一个状态机, 其输入界面图如图 1-3 所示。

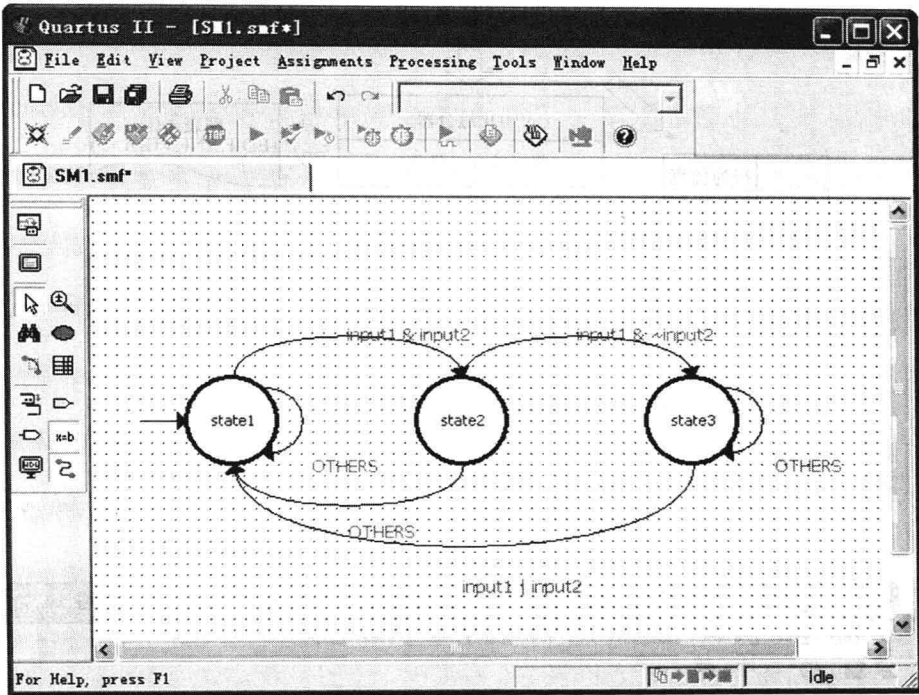


图 1-3 状态机输入界面图

这种图形状态机设计方法中, 设计者不必关心 PLD 内部结构和布尔表达式, 只需要考虑状态转移条件及各状态之间关系, 使用作图方法构成状态转移图, 由计算机自动化生成 VHDL 或其他形式的语言描述的功能模块。

(4) 波形输入法 对于那些只关心输入与输出信号之间的关系, 而不需要对中间变量进行干预的系统可使用波形输入法。该方法只需给出输入信号与输出信号的波形, 主要用建立和编辑波形设计文件及仿真向量和功能测试向量。波形设计输入系统可以根据用户定义的输入/输出波形自动生成逻辑关系。EDA 软件会自动生成相应功能模块, 其语言可由设计者选择。波形输入法是一种简明的设计方法, 并且容易查错。该方法编译软件复杂, 不适合复杂系统设计, 只有在少数 EDA 软件中有集成。

2. 逻辑综合和优化

欲把 VHDL 的软件设计与硬件的可实现性挂钩, 需要利用 EDA 软件系统的综合器进行逻辑综合。

综合器的功能就是将设计者在 EDA 平台上完成的针对某个系统项目的 VHDL 程序、原

理图或状态图形的描述，针对给定硬件结构组件进行编译、优化、转换和综合，最终获得门级电路甚至更底层的电路描述文件。由此可见，综合器工作前，必须给定最后实现的硬件结构参数，它的功能就是将软件描述与给定硬件结构用某种图表文件的方式联系起来。显然，综合器是软件描述与硬件实现的一座桥梁。综合过程就是将电路的高级语言描述转换成低级的，可与 FPGA/CPLD 或构成 ASIC 的门阵列基本结构相映射的网表文件。

由于 VHDL 仿真器的行为仿真功能是面向高层次的系统仿真，只能对 VHDL 的系统描述做可行性的评估测试，不针对任何硬件系统，因此基于这一仿真层次的许多 VHDL 语句不能被综合器所接受。这就是说，这类语句的描述无法在硬件系统中实现（至少是现阶段），这时，综合器不支持的语句在综合过程中将忽略掉。综合器对源 VHDL 文件的综合是针对某一 PLD 供应商的产品系列的，因此，综合后的结果是可以为硬件系统所接受，具有硬件可实现性。

3. 目标器件的布线/适配

逻辑综合通过后必须利用适配器将综合后的网表文件针对某一具体的目标器进行逻辑映射操作，其中包括底层器件配置、逻辑分割、逻辑优化、布线与操作等，配置于指定的目标器件中，产生最终的下载文件，如 JEDEC 格式的文件。

适配所选定的目标器件（FPGA/CPLD 芯片）必须属于原综合器指定的目标器件系列。对于一般的可编程模拟器件所对应的 EDA 软件来说，一般仅需包含一个适配器就可以了，如 Lattice 公司的 PAC-DESIGNER。通常，EDA 软件中的综合器可由专业的第三方 EDA 公司提供，而适配器则需由 FPGA/CPLD 供应商自己提供，因为适配器的适配对象直接与器件结构相对应。

4. 目标器件的编程/下载

如果编译、综合、布线/适配和行为仿真、功能仿真、时序仿真等过程都没有发现问题，即满足原设计的要求，则可以将由 FPGA/CPLD 布线/适配器产生的配置下载文件通过编程器或下载电缆载入目标芯片 FPGA 或 CPLD 中。

5. 设计过程中的有关仿真

设计过程中的仿真有 3 种，它们是行为仿真、功能仿真和时序仿真。

所谓行为仿真，就是将 VHDL 设计源程序直接送到 VHDL 仿真器中进行的仿真。该仿真是根据 VHDL 的语义进行的，与具体电路没有关系。在这种仿真中，可以充分发挥 VHDL 中适用于仿真控制的语句及有关的预定义函数和库文件。

所谓功能仿真，就是将综合后的 VHDL 网表文件再送到 VHDL 仿真器中所进行的仿真。这时功能仿真是仅对 VHDL 描述的逻辑功能进行测试模拟，以了解其实现的功能是否满足原设计的要求，仿真过程不涉及具体器件的硬件特性，如延时特性。该仿真结果与门级仿真器所做的功能仿真的结果基本一致。

所谓时序仿真，就是将布线/适配器所产生的 VHDL 网表文件送到 VHDL 仿真器所进行的仿真。仿真过程中已将器件特性考虑进去了，因而可以得到精确的时序仿真结果。通过布线/适配的处理后，布线/适配器将生成一个 VHDL 网表文件，这个网表文件中包含了较为精确的延时信息，网表文件中描述的电路结构与布线/适配后的结果是一致的。

需要注意的是，图 1-1 中有两个仿真器：一是 VHDL 仿真器，另一个是门级仿真器，它们都能进行功能仿真和时序仿真。所不同的是仿真用的文件格式不同，即网表文件不同。这

里所谓的网表 (Net list), 是特指电路网络, 网表文件描述了一个电路网络。目前流行多种网表文件格式, 其中最通用的是 EDIF 格式的网表文件。VHDL 文件格式也可以用来描述电路网络, 即采用 VHDL 语法描述各级电路互连, 称之为 VHDL 网表。

6. 硬件仿真/硬件测试

所谓硬件仿真是针对 ASIC 设计而言的。常利用 FPGA 对系统的设计进行功能检测, 通过后再将其 VHDL 设计以 ASIC 形式实现。所谓硬件测试就是 FPGA/CPLD 直接用于应用系统的设计中, 将下载文件下载到目标器件后, 对系统的设计进行的功能检测的过程。

硬件仿真和硬件测试的目的, 是为了在更真实的环境中检验 VHDL 设计的运行情况, 特别是对于 VHDL 程序设计上不是十分规范、语义上含有一定歧义的程序。一般的仿真器包括 VHDL 行为仿真器和 VHDL 功能仿真器, 它们对于同一 VHDL 设计的“理解”, 即仿真模型的产生, 与 VHDL 综合器的“理解”, 即综合模型的产生, 常常是不一致的。此外, 由于目标器件功能的可行性约束, 综合器对于设计的“理解”常在一个有限范围内选择, 而 VHDL 仿真器的“理解”是纯软件行为, 其“理解”的选择范围要宽得多, 结果这种“理解”的偏差势必导致仿真结果与综合后实现的硬件电路在功能上的不一致。当然, 还有许多其他的因素也会产生这种不一致, 由此可见, VHDL 设计的硬件仿真和硬件测试是十分必要的。

1.6 数字系统的设计

1.6.1 数字系统的设计模型

数字系统指的是交互式的、以离散形式表示的具有存储、传输、信息处理能力的逻辑子系统的集合。用于描述数字系统的模型有多种, 各种模型描述数字系统的侧重点不同。下面介绍一种普遍采用的模型。这种模型根据数字系统的定义, 将整个系统划分为两个模块或两个子系统: 数据处理子系统和控制子系统, 如图 1-4 所示。

数据处理子系统主要完成数据的采集、存储、运算和传输。数据处理子系统主要由存储器、运算器、数据选择器等功能电路组成。数据处理子系统与外界进行数据交换, 在控制子系统 (控制器) 发出的控制信号作用下, 数据处理子系统将进行数据的存储和运算等操作。数据处理子系统将接收由控制器发出的控制信号, 同时将自己的操作进程或操作结果作为条件信号传给控制器。应当根据数字系统实现的功能或算法设计数据处理子系统。

控制子系统是执行数字系统算法的核心, 具有记忆功能, 因此控制子系统是时序系统。控制子系统由组合逻辑电路和触发器组成, 与数据处理子系统共用时钟。控制子系统的输入信号是外部控制信号和由数据处理子系统送来的条件信号, 按照数字系统设计方案要求的算法流程, 在时钟信号的控制下进行状态的转换, 同时产生与状态和条件信号相对应的输出信号, 该输出信号将控制数据处理子系统的操作。应当根据数字系统功能及数据处理子系统的需求设计控制子系统。

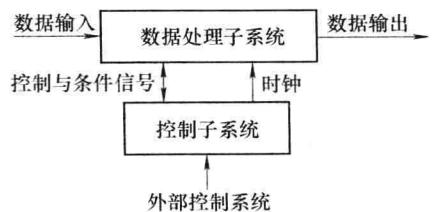


图 1-4 数字系统的设计模型

把数字系统划分成数据处理子系统和控制子系统进行设计,只是一种手段,不是目的。它用来帮助设计者有层次地理解和处理问题,进而获得清晰、完整正确的电路图。因此,数字系统的划分应当遵循自然、易于理解的原则。

设计一个数字系统时,采用该模型的优点是:

1) 把数字系统划分为控制子系统和数据处理子系统两个主要部分,使设计者面对的电路规模减小,二者可以分别设计。

2) 由于数字系统中控制子系统的逻辑关系比较复杂,将其独立划分出来后,可突出设计重点和分散设计难点。

3) 当数字系统划分为控制子系统和数据处理子系统后,逻辑分工清楚,各自的任务明确,这可以使电路的设计、调试测量和故障处理都比较方便。

但采用该模型设计一个数字系统时,必须先分析和找出实现系统逻辑的算法,根据具体的算法要求提出系统内部的结构要求,再根据各个部分分担的任务划分出控制子系统和数据处理子系统。算法不同,系统的内部结构不同,控制子系统和数据处理子系统电路也不同。有时控制子系统和数据处理子系统的界限划分也比较困难,需要反复比较和调整才能确定。

1.6.2 数字系统的设计方法

数字系统设计有多种方法,如模块设计法、自顶向下设计法和自底向上设计法等。

数字系统的设计一般采用自顶向下、由粗到细、逐步求精的方法。自顶向下是指将数字系统的整体逐步分解为各个子系统和模块,若子系统规模较大,则还需将子系统进一步分解为更小的子系统和模块,层层分解,直至整个系统中各子系统关系合理,并便于逻辑电路级的设计和实现为止。采用该方法设计时,高层设计进行功能和接口描述,说明模块的功能和接口,模块功能的更详细的描述在下一设计层次说明,最底层的设计才涉及具体的寄存器和逻辑门电路等实现方式的描述。

采用自顶向下的设计方法有如下优点:

1) 自顶向下设计方法是一种模块化设计方法。对设计的描述从上到下逐步由粗略到详细,符合常规的逻辑思维习惯。由于高层设计与器件无关,设计易于在各种集成电路工艺或可编程器件之间移植。

2) 适合多个设计者同时进行设计。随着技术的不断进步,许多设计由一个设计者已无法完成,必须经过多个设计者分工协作完成一项设计的情况越来越多。在这种情况下,应用自顶向下的设计方法便于由多个设计者同时进行设计,对设计任务进行合理分配,用系统工程的方法对设计进行管理。

针对具体的设计,实施自顶向下的设计方法的形式会有所不同,但均需遵循以下两条原则:逐层分解功能,分层次进行设计。同时,应在各个设计层次上,考虑相应的仿真验证问题。

1.6.3 数字系统的设计准则

进行数字系统设计时,通常需要考虑多方面的条件和要求,如设计的功能和性能要求,元器件的资源分配和设计工具的可实现性,系统的开发费用和成本等。虽然具体设计的条件和要求千差万别,实现的方法也各不相同,但数字系统设计还是具备一些共同的方法和准则

的。

1. 分割准则

自顶向下的设计方法或其他层次化的设计方法，需要对系统功能进行分割，然后用逻辑语言进行描述。分割过程中，若分割过粗，则不易用逻辑语言表达；分割过细，则带来不必要的重复和繁琐。因此，分割的粗细需要根据具体的设计和设计工具情况而定。掌握分割程度，可以遵循以下原则：分割后最底层的逻辑块应适用逻辑语言进行表达；相似的功能应该设计成共享的基本模块；接口信号尽可能少；同层次的模块之间，在资源和 I/O 分配下，尽可能平衡，以便结构匀称；模块的划分和设计，尽可能做到通用性好，易于移植。

2. 系统的可观测性

在系统设计中，应该同时考虑功能检查和性能的测试，即系统可观测性的问题。一些有经验的设计者会自觉地在设计系统的同时设计观测电路，即观测器，指示系统内部的工作状态。

建立观测器，应遵循以下原则：具有系统的关键点信号，如时钟、同步信号和状态等信号；具有代表性的节点和线路上的信号；具备简单的“系统工作是否正常”的判断能力。

3. 同步和异步电路

异步电路会造成较大延时和逻辑竞争，容易引起系统的不稳定，而同步电路则是按照统一的时钟工作，稳定性好。因此在设计时应尽可能采用同步电路进行设计，避免使用异步电路。在必须使用异步电路时，应采取避免竞争和增加稳定性。

4. 最优化设计

由于可编程器件的逻辑资源、连接资源和 I/O 资源有限，器件的速度和性能也是有限的，用器件设计系统的过程相当于求最优解的过程。因此，需要给定两个约束条件：边界条件和最优化目标。

所谓边界条件，是指器件的资源及性能限制。最优化目标有多种，设计中常见的最优化目标有：器件资源利用率最高；系统工作速度最快，即延时最小；布线最容易，即可实现性最好。具体设计中，各个最优化目标间可能会产生冲突，这时应满足设计的主要要求。

5. 系统设计的艺术

一个系统的设计，通常需要经过反复地修改、优化才能达到设计的要求。一个好的设计应该满足“和谐”的基本特征，对数字系统可以根据以下几点做出判断：

设计是否总体上流畅，无拖泥带水的感觉；资源分配、I/O 分配是否合理，设计上和性能上的是否有瓶颈，系统结构是否协调；是否具有有良好的可观测性；是否易于修改和移植；器件的特点是否能得到充分的发挥。

1.6.4 数字系统的设计步骤

1. 系统任务分析

数字系统设计中的第一步是明确系统的任务。在设计任务书中，可用各种方式提出对整个数字系统的逻辑要求，常用的方式有自然语言、逻辑流程图、时序图或几种方法的结合。当系统较大或逻辑关系较复杂时，系统任务（逻辑要求）逻辑的表述和理解都不是一件容易的工作。所以，分析系统的任务必须细致、全面，不能有理解上的偏差和疏漏。