

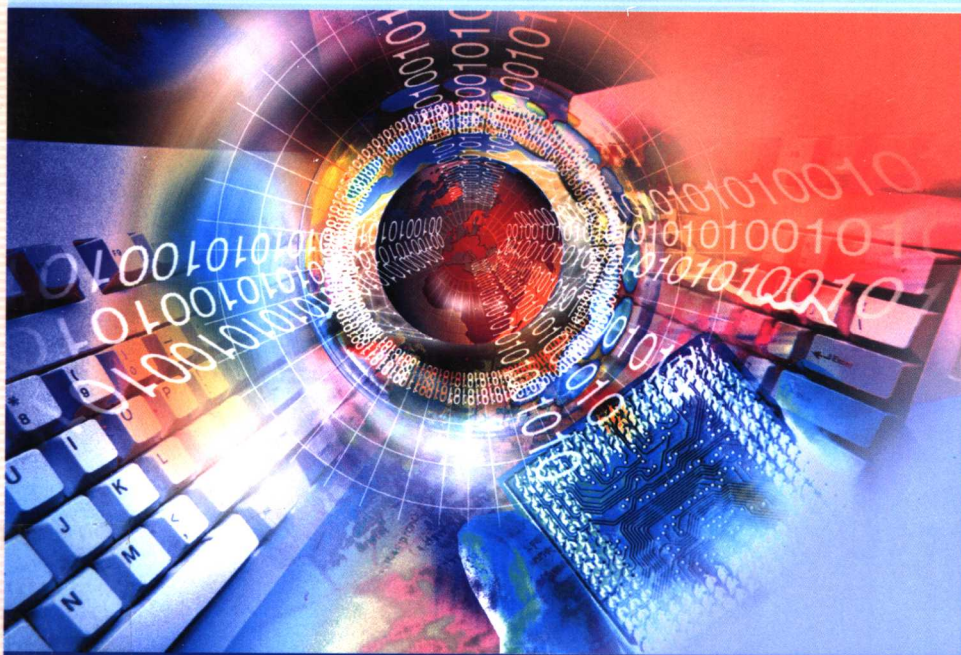


新世纪电子信息平台课程系列教材

数字系统设计及VHDL实践

徐向民 主编

SHUZI XITONG SHEJI JI VHDL SHIJIAN



TP271/79

2007

新世纪电子信息平台课程系列教材

数字系统设计及 VHDL 实践

主编 徐向民

参编 邢晓芬 邓洪波 李 磊

机械工业出版社

本书系统地介绍了数字系统描述的硬件语言与设计方法,从理论、方法、工具,到实践进行了全面的阐述。全书分两篇:基础篇和实践篇。基础篇,共有8章。第1章介绍了数字系统的基本概念及EDA技术的发展趋势;第2、3章结合大量实例,介绍了硬件描述语言VHDL;第4、5章介绍了基于ASM图的时序电路设计方法及状态机的VHDL实现;第6、7章介绍了有关仿真和综合的知识;第8章结合设计实例,介绍了数字系统设计方法的原理和具体的应用。实践篇共有5章,从实践的角度,由浅入深,结合自主开发的EDA实验平台,从设计描述到下载对数字系统的整个设计流程进行了详细的讲解。本书兼具知识性和实用性。

本书可作为大专院校电子类高年级本科生和研究生学习数字系统设计的教科书和参考书,也可作为这一领域工程技术人员的参考书。

图书在版编目(CIP)数据

数字系统设计及VHDL实践/徐向民主编. —北京:机械工业出版社,2007.10
(新世纪电子信息平台课程系列教材)
ISBN 978-7-111-22506-5

I. 数… II. 徐… III. ①数字系统-系统设计-教材
②硬件描述语言, VHDL-教材 IV. TP271 TP312

中国版本图书馆CIP数据核字(2007)第153154号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
责任编辑:贡克勤 责任校对:吴美英
封面设计:鞠 杨 责任印制:邓 博
北京京丰印刷厂印刷
2007年10月第1版·第1次印刷
184mm×260mm·14.75印张·359千字
标准书号:ISBN 978-7-111-22506-5
定价:22.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话:(010) 68326294
购书热线电话:(010) 88379639 88379641 88379643
编辑热线电话:(010) 88379711
封面无防伪标均为盗版

新世纪电子信息平台课程系列教材

编 委 会

主任委员	徐向民				
副主任委员	殷瑞祥	贡克勤			
委 员	褚庆昕	冯穗力	傅予力	金连文	
	林土胜	陆以勤	丘水生	尹俊勋	

前 言

随着微电子技术的发展,设计与制造集成电路的任务已不完全由半导体厂商独立承担。可编程逻辑器件的出现,给数字系统的设计方法带来了革命性的变化,它为用户最终把自己所设计的逻辑电路直接写入到芯片上提供了物质基础,大大减轻了电路图设计和电路板设计的工作量和难度,从而有效地增强了设计的灵活性,提高了工作效率。

为了全面地阐述清楚基于可编程器件的现代数字系统设计方法,作者在参考国内外出版的有关 EDA 技术、VHDL 方面书的基础上,结合多年的理论教学、实践教学及相关科研经验,编写了这本书。本书最大的特点是内容全面,突出设计方法的应用和工程实践。全书共分两篇:基础篇和实践篇。基础篇共有 8 章,包括硬件描述语言 VHDL 基础、时序电路设计方法、状态机的 VHDL 实现、仿真、综合及数字系统设计方法的原理。实践篇共有 5 章,介绍了可编程逻辑器件、基于 CPLD/FPGA 的数字系统设计,通过简单的程序和实际工程项目的简化例子,由浅入深,实践了硬件描述语言和数字系统设计方法,所有实际工程例子均在自主开发的 EDA 实验平台上得到验证。

现代电子设计已经进入数字化时代,社会对 EDA 方面的人才需求越来越大。本书的目的是引导电子类的高校学生建立自顶向下的现代数字系统设计理念,从传统搭积木式的自底向上设计转向基于可编程逻辑器件的数字系统设计。

本书由徐向民任主编,邢晓芬、邓洪波、李磊参编。特别感谢吴淑泉老师给予的指导,同时感谢梁惠华、蔡舒贝、邝友和、郭振灵、黄国辉等同学付出的艰辛劳动。本书承华南理工大学电子与信息学院教学指导委员会的审阅,对本书的编写提出了宝贵的意见,在此表示衷心的感谢。另外在书中用到的部分资料来自于同行的积累,在此一并表现感谢,在参考文献中若有遗漏,请批评指正。

本书在编写过程中,着眼设计方法与实践的结合,力尽做到全面、易读、深入浅出。由于编者自身的水平有限,如果书中存在错误和不妥之处,敬请读者批评指正。

编 者

目 录

前言

基 础 篇

第 1 章 数字系统设计与 EDA

技术 1

- 1.1 数字系统概念 1
- 1.2 电子设计发展趋势 3
- 1.3 EDA 技术介绍 4
 - 1.3.1 基本特征 4
 - 1.3.2 主要内容 4
 - 1.3.3 EDA 设计流程 6
- 1.4 EDA 应用与发展趋势 7

第 2 章 VHDL 语言基础 8

- 2.1 硬件描述语言特点 8
- 2.2 VHDL 程序基本结构 8
- 2.3 VHDL 程序主要构件 10
 - 2.3.1 库 10
 - 2.3.2 实体 10
 - 2.3.3 结构体 11
- 2.4 VHDL 数据类型 12
 - 2.4.1 标准数据类型 12
 - 2.4.2 用户自定义数据类型 14
 - 2.4.3 数据类型转换 14
- 2.5 运算符 15
 - 2.5.1 算术运算符 15
 - 2.5.2 逻辑运算符 15
 - 2.5.3 关系运算符 16
 - 2.5.4 其他运算符 16
 - 2.5.5 运算优先级 16
- 2.6 VHDL 数据对象 16
 - 2.6.1 常量 16
 - 2.6.2 变量 17
 - 2.6.3 信号 18
 - 2.6.4 信号与变量的比较 19
- 2.7 VHDL 基本语句 19
 - 2.7.1 并行语句 20

- 2.7.2 顺序语句 27
- 2.7.3 属性描述语句 33
- 2.8 测试基准 35
- 2.9 VHDL 程序的其他构件 36
 - 2.9.1 块 36
 - 2.9.2 函数 37
 - 2.9.3 过程 39
 - 2.9.4 程序包 40
- 2.10 结构体的描述方法 42

第 3 章 组合逻辑模块 43

- 3.1 简单组合逻辑模块 43
- 3.2 译码器 44
- 3.3 优先级编码器 45
- 3.4 补码器 45
- 3.5 三态门 46
- 3.6 总线缓冲器 46
- 3.7 多路选择器 47
- 3.8 全加器 48
- 3.9 串行进位加法器 49
- 3.10 并行进位加法器 50
- 3.11 比较器 51
- 3.12 只读存储器 52
- 3.13 随机存储器 53

第 4 章 同步时序电路设计 55

- 4.1 时序电路的特点与组成 55
- 4.2 设计举例——3 位计数器 56
- 4.3 时序电路描述方法 58
 - 4.3.1 ASM 图的组成 59
 - 4.3.2 自动售邮票机 60
 - 4.3.3 状态分配与编码 62
 - 4.3.4 状态最少化 62
- 4.4 ASM 图的硬件实现 63



4.4.1 计数器法	64	6.1 仿真的级别	98
4.4.2 多路选择器法	65	6.2 逻辑仿真	98
4.4.3 定序法	67	6.3 延时模型	99
4.4.4 微程序法	68		
4.5 有限状态机的 VHDL 实现	69	第 7 章 综合	101
4.5.1 符号化状态机	69	7.1 综合的层次	101
4.5.2 单进程状态机	71	7.2 高层次综合	102
4.5.3 双进程状态机	73	7.3 寄存器传输级综合	102
4.5.4 三进程状态机	75	7.3.1 不能综合的 VHDL 描述	103
4.5.5 设计实例 1——序列检测器	78	7.3.2 寄存器的引入方法	103
4.5.6 设计实例 2——A/D 采样控 制器	79	7.3.3 避免引入不必要的寄存器	111
4.6 关联状态机	83	7.4 约束条件	117
4.7 数字系统控制器/数据处理器 模型	84	7.5 可编程器件综合	119
第 5 章 基本时序逻辑电路	86	第 8 章 数字系统设计方法	121
5.1 锁存器	86	8.1 数字系统层次化设计	121
5.1.1 RS 锁存器	86	8.1.1 数字系统层次化结构	121
5.1.2 D 锁存器	87	8.1.2 自顶向下设计方法	122
5.2 触发器	87	8.2 模块划分技术	122
5.2.1 D 触发器	87	8.3 设计实例——串行数据接收器	122
5.2.2 带有 $\bar{Q}$ 输出的 D 触发器	89	8.3.1 性能级设计	123
5.2.3 JK 触发器	91	8.3.2 系统结构级设计	123
5.2.4 T 触发器	92	8.3.3 逻辑级设计	124
5.3 多位寄存器	93	8.3.4 利用 VHDL 简化逻辑级设计	126
5.4 串进并出型移位寄存器	93	8.3.5 物理级设计	130
5.5 计数器	94	8.4 迭代技术	130
5.6 无符号数乘法器	96	8.4.1 空间迭代	130
		8.4.2 时间迭代	132
		8.4.3 二维迭代	133
第 6 章 仿真	98		
实 践 篇			
第 9 章 可编程逻辑器件	135	10.1 基于 Quartus II 的数字系统设 计流程	142
9.1 可编程逻辑器件的发展	135	10.2 基于 CPLD/FPGA 的 EDA 实验平台 设计	143
9.2 PLD 的分类	136	10.3 Quartus II 软件使用介绍	143
9.2.1 基于乘积项技术的 PLD	136		
9.2.2 基于查找表技术的 PLD	139	第 11 章 组合逻辑电路实验	164
9.3 CPLD 与 FPGA 的比较	141	11.1 4 选 1 多路选择器	164
第 10 章 基于 CPLD/FPGA 的数字 系统设计	142	11.2 16 位加法器设计	165

第 12 章 时序电路实验	167	13.2.3 系统的设计与实现	193
第 13 章 综合性设计实验	172	13.2.4 波形仿真与分析	199
13.1 三人抢答器	172	13.2.5 思考题	201
13.1.1 设计要求	172	13.3 红外遥控器	202
13.1.2 设计分析与设计思路	172	13.3.1 设计要求	202
13.1.3 各模块的设计与实现	174	13.3.2 设计分析与设计思路	202
13.1.4 整体设计	182	13.3.3 各模块的设计与实现	207
13.1.5 波形仿真与分析	185	13.3.4 整体设计	219
13.1.6 思考题	190	13.3.5 波形仿真与分析	219
13.2 出租车计费实验	191	13.3.6 硬件配置	223
13.2.1 设计要求	191	13.3.7 思考题	224
13.2.2 设计分析与设计思路	191	参考文献	225

基 础 篇

第 1 章 数字系统设计与 EDA 技术

传统的数字系统设计是基于电路板的，需选用大量的固定功能器件，再通过器件的配合，设计模拟系统功能，工作集中在器件的选用及电路板的设计上。随着计算机性价比的提高及可编程逻辑器件的出现，现代数字系统的设计，设计师可以通过设计芯片来实现电子系统的功能，将传统的固件选用及电路板设计工作放在芯片设计中进行。

20 世纪 90 年代初开始，电子产品设计系统日趋数字化复杂化和大规模集成化各种电子系统的设计软件应运而生。其中，EDA（电子设计自动化）有一定的代表性。它是基于芯片的，优势在于能用 HDL（硬件描述）语言进行输入、进行 PLD（可编程器件）的设计与仿真等系统设计自动化。EDA 技术一出现就显示出极大的优势，现在已经成为数字系统设计的主流技术。

1.1 数字系统概念

在电子技术飞速发展的今天，人类正跨入信息时代。从计算机到 GSM 移动电话，从家用娱乐使用的 VCD、HDTV 数字电视到军用雷达、医用 CT 仪器等设备，数字化技术比比皆是，涉及通信、国防、航天、医学、工业自化、计算机应用、仪器仪表等领域。数字系统的使用已经成为构成现代电子系统的重要标志。

1. 基本概念

数字系统是指对数字信息进行储存、传输、处理的电子系统，它的输入和输出都是数字量。在结构上分为数据处理单元和控制单元，如图 1-1 所示。

符合上述结构的系统都可以看作数字系统，从目前来看，数字系统的实现可以有多种方式，早期的数字电路是用中小规模元件构成。随着数字系统复杂度增加，中小规模元件已经很难实现，芯片技术的发展提供了两种可能：微处理器（Micro Process Unit，MPU）和可编程逻辑器件（Programmable Logic Device，PLD）。

微处理器是具有运算器和控制器功能的大规模集成电路的芯片，主要包括：嵌入式、单片机、DSP，具有体积小、开发方便、成本低的特点，但微处理器是通过执行程序来实现逻辑功能控制，所以速度较慢；可编程逻辑器件虽然硬件成本和开发门槛较高，但因为用可编程逻辑实现的是硬件电路，所以非常适合用于需高速处理的应用环境，如 3G 通信的基带处理。并且数字系统用可编程逻辑器件实现后，很容易过渡到专用芯片，这样可以大大缩短芯片的开发周期和开发成本。

随着系统复杂性进一步提高以及成本的考虑，现在很多数字系统的实现往往既用了微处

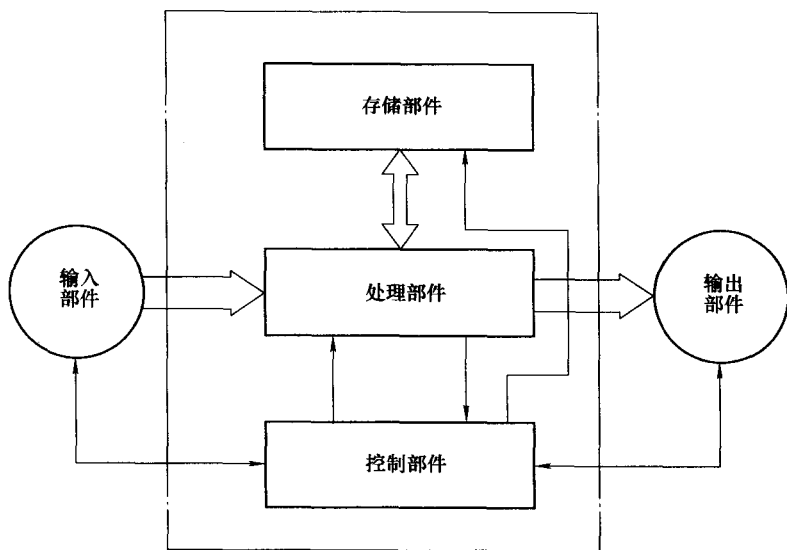


图 1-1 数字系统结构

理器又用了可编程逻辑器件，而且现在可编程逻辑器件芯片容量越来越大，在可编程逻辑器件芯片中实现微处理器也成为可能（例：Altium 奥腾有限公司对 FPGA 设计提供了丰富的 IP 内核，包括各种处理器、存储器、外设、接口以及虚拟仪器）。因篇幅所限，本书只讨论用 VHDL 语言在常规的可编程逻辑器件设计数字系统的理论和方法。

2. 设计方法

传统的数字系统设计是自下向上的设计方法，首先确定系统最底层的电路模块或元件的结构和功能，建立相应的数学模型，用数值计算各项参数，与设计目标反复比较过程中修改或完善模型，按要求写出输入、输出表达式或状态图，用真值表、卡诺图化简。然后根据主系统的功能要求，将它们组合成更大的功能模块，直到完成整个目标系统的设计。因此，只有在设计完成后才能进行仿真，存在的问题才能被发现。

同时，在系统进行细分时，必须考虑现有并能获得的器件（往往是标准的器件），而且必须对各种具体器件功能、性能指标及连接方式非常熟悉。设计者往往需要较长时间的训练和经验积累，采用试凑的方法才能设计出满足要求的数字系统，有时甚至达不到系统设计的某些要求，所以适用于小规模集成电路系统设计。

现代数字系统设计可以直接面向用户需求，根据系统的行为和功能要求，自上至下地逐层完成相应的描述、综合、优化、仿真与验证，直到把设计结果下载到器件中。上述设计过程除了系统行为和功能描述以外，其余所有的设计过程几乎都可以用计算机来自动地完成，即电子设计自动化（EDA）。这些设计方法大大地缩短了系统的设计周期，适应当今电子市场品种多、批量小的需求，提高了产品的竞争能力。设计步骤如图 1-2 所示。

图 1-2 中，行为设计确定系统的功能、性能及允许的芯片面积和成本等；结构设计根据系统或芯片特点，将其分解为接口清晰、关系明确、尽可能简单的子系统，包括算术运算单元、控制单元、数据通道等；逻辑设计把结构转换成逻辑图，尽可能采用规则的单元或模块；电路设计将逻辑图转换成电路图，需仿真确定逻辑图正确性。版图设计即芯片设计，把

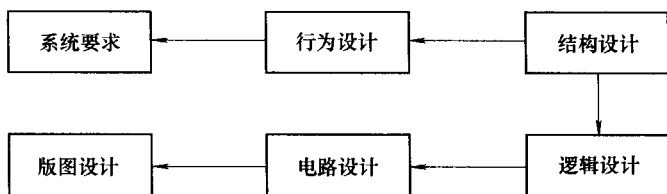


图 1-2 自上而下的设计步骤

设计完的电路集成到具体芯片中。

1.2 电子设计发展趋势

电子产品发展迅速，向着功能多样化、体积最小化、功耗最低化的方向前进，同时价格也呈下降趋势，主要原因是生产制造技术和电子设计技术的发展。前者以微电子加工技术为代表，已经达到了深亚微米的工艺水平，可以在几平方厘米的芯片上集成数千万个晶体管；后者核心是电子设计自动化（Electronic Design Automation, EDA）技术。

根据电子设计的发展特征，EDA 发展可分为三个阶段：

20 世纪 70 年代 CAD（Computer Assist Design，计算机辅助设计）的诞生，用计算机设计印制电路板（Printed Circuit Board, PCB）取代纯手工操作，设计简单易行，但成本高，功能有限，易仿制，可制造性差，数据不可重复。

20 世纪 80 年代 CAM（计算机辅助制造）、CAT（计算机辅助测试）、CAE（计算机辅助工程）的产生，主要用于电器原理图的输入、逻辑仿真、电路分析、布局布线和 PCB 设计。编程灵活，可制造性好，可重复用数据。缺点是开发成本高，保密性差，不适用于高速和实时处理系统的应用。

20 世纪 90 年代开始，各类可编程半导体芯片的生产推动了芯片设计技术的发展，硬件描述语言的产生和完善使得大规模专用集成电路的设计和仿真得到保证，包括算法设计、芯片设计和电路设计。其开发周期短，保密性好，系统的总成本低。

集成电路设计和工艺技术水平大大提高，单片集成度中每片已能含上亿个晶体管，使得将原来由许多 IC 组成的电子系统集成在一个单片硅片上成为可能，构成所谓的片上系统（System on Chip, SOC）。它将信号采集、处理和输入输出等完整的系统功能集成在一起，成为一个专用功能的电子系统芯片。随后出现了 SOPC（片上可编程系统），它是用 PLD 取代 ASIC 的更灵活、高效的 SOC 技术，特点在于可编程性，所设计的电路系统在规模、可靠性、体积、功能、开发成本等方面实现最优化。

例如，在 2000 年，Altera 发布了 Nios 处理器，这是 Altera Excalibur 嵌入处理器计划中第一个产品，它成为业界第一款为可编程逻辑优化的可配置处理器。Altera 很清楚地意识到，如果把可编程逻辑的固有的优势集成到嵌入处理器的开发流程中，就会拥有非常成功的产品。一旦定义了处理器之后，设计者就“具备”了体系结构，可马上开始设计软件原型。CPU 周边的专用硬件逻辑可以慢慢地集成进去，在每个阶段软件都能够进行测试，解决问题。这款芯片集成了传统的可编程逻辑器件的优势，又融合了微处理器的优点，能真正实现



SOPC 设计。

1.3 EDA 技术介绍

1.3.1 基本特征

电子设计自动化 (EDA) 汇集了计算机应用科学、电子系统科学、微电子科学等多学科内容, 它以计算机为工具, 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, 以专用集成电路 (Application Specific Integrated Circuits, ASIC)、片上系统 (SOC) 芯片为目标器件, 自动完成逻辑编译、逻辑化简、逻辑综合、结构综合 (布局布线), 以及逻辑优化和仿真测试, 直至实现既定的电子系统功能。

利用 EDA 设计数字系统具有以下几个特点: 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; ③采用自顶向下的设计方法; ④设计过程中可用有关软件进行各种仿真; ⑤系统可现场编程, 在线升级; ⑥整个系统可集成在一个芯片上, 体积小、功耗低、可靠性高。

1.3.2 主要内容

1. 大规模可编程逻辑器件

可编程逻辑器件 (Programmable Logic Device, PLD) 可直接从市场上购得, 用户只要通过对器件编程就可实现所需要的逻辑功能。这种设计方法成本低、使用灵活、设计周期短、可靠性高、承担风险小。可编程逻辑器件发展到现在, 规模越来越大, 功能越来越强, 价格越来越低, 相配套的 EDA 软件越来越完善, 因而受到广大设计人员的喜爱。目前, 在电子系统开发阶段的硬件验证过程中, 一般都采用可编程逻辑器件, 以期尽快开发新产品。

随着可编程逻辑器件应用的日益广泛, 许多 IC 制造厂家涉足 PLD/FPGA 领域。目前世界上有十几家生产 CPLD/FPGA 的公司, 最大的三家是: Altera, Xilinx, Lattice, 其中 Altera 和 Xilinx 占有了 60% 以上的市场份额。

例如, 2004 年底 Altera 公司开发了新一代 PLD 器件 MAXII, 采用 FPGA 结构, 配置芯片集成在内部, 和普通 PLD 一样上电即可工作。容量比上一代大大增加, 内部集成一片 8Kbit 串行 EEPROM, 增加很多功能。不过 MAXII 容量较大, 对于只需要几十个逻辑单元的简单逻辑应用, 建议使用小容量的 EPM3000A 系列芯片。

随着微电子技术的发展, 设计与制造集成电路的任务已不完全由半导体厂商来独立承担。系统设计师更愿意自己设计专用集成电路 (ASIC) 芯片, 而且希望 ASIC 的设计周期尽可能短, 最好是在实验室里就能设计出合适的 ASIC 芯片, 并且立即投入实际应用之中, 因而出现了现场可编程逻辑器件 (Field Programmable Logic Device, FPLD), 其中应用最广泛的当属现场可编程门阵列 (Field programmable Gate Array, FPGA) 和复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD)。

例如, Altera 公司的主流 FPGA 分为两大类: 一类侧重低成本应用, 容量中等, 性能可以满足一般的逻辑设计要求, 如 Cyclone、CycloneII; 还有一类侧重于高性能应用, 容量大, 性能能满足各类高端应用, 如 Startix、StratixII 等, 用户可以根据自己实际应用要求进行选



择。在性能可以满足的情况下，优先选择低成本器件。Xilinx 的主流 FPGA 也分为两大类，代表有产品分别为 Spartan 系列和 Virtex 系列，其中 Spartan 系列主要应用于低成本设计，Virtex 系列主要应用于高端设计。

Lattice 在 PLD 领域发展多年，拥有众多产品系列，目前主流产品是 ispMACH4000、MachXO 系列 CPLD 和 LatticeEC/ECP 系列 FPGA，此外，在混合信号芯片上，也有诸多建树，如可编程模拟芯片 ispPAC、可编程电源管理、时钟管理等。

当前电子产品市场需求以及生产制造技术的不断进步，标志着 EDA 技术发展成果的最新器件不断涌现，并且向着大规模、低功耗、多功能方向发展。例如，采用系统级性能复杂可编程逻辑技术（CPLD）和现场可编程门阵列（FPGA）实现可编程 SOC 已成为今后的一个发展方向。

2. 硬件描述语言

EDA 技术中多采用硬件描述语言（Hardware Description Language，HDL）描述电子系统的逻辑功能、电路结构和连接形式。HDL 可以在三个层次上进行电路描述，由高到低为系统行为级、寄存器传输级和逻辑门级，支持结构、数据流、行为三种描述形式的混合描述。常用的 HDL 有 VHDL、Verilog 和 AHDL 语言。VHDL 适用于行为级和寄存器传输级；Verilog 和 AHDL 适用于寄存器级和门电路级。现在 VHDL 和 Verilog 作为工业标准硬件描述语言，在电子工程领域已成为通用的 HDL，承担几乎全部的数字系统的设计任务。

用 VHDL 设计电子系统有以下优点：①更强的行为描述能力，避开具体的器件结构，从逻辑行为上描述和设计大规模电子系统；②具有丰富的仿真语句和库函数，使得在设计早期就能检查设计系统的功能可行性，并可以随时对系统仿真；③用 VHDL 完成的设计，可以用 EDA 工具进行逻辑综合和优化，根据不同的目标芯片自动把描述设计转化成门级网表，从而极大地减少了设计时间和可能发生的错误；④设计描述有相对独立性，可以在不懂硬件结构情况下进行设计；⑤可以在不改变源程序前提下，只改变类属参量或函数，就能很容易改变设计规模和结构。

3. 软件开发工具

集成的 PLD/FPGA 开发软件见表 1-1，这类软件一般由芯片厂家提供，基本可以完成所有设计输入、仿真、综合、布线、下载等工作。

表 1-1 集成的 PLD/FPGA 开发软件

供应商	开发软件	简介
Altera	Maxplus II	曾经是最优秀的 PLD 开发平台之一，适合开发早期的中小规模 PLD/FPGA，使用者众多。
	Quartus II	新一代 PLD 开发软件，适合大规模 FPGA 的开发
	SOPC Builder	配合 Quartus II，完成集成 CPU 的 FPGA 芯片的开发工作
	DSP Builder	Quartus II 与 Mathlab 的接口，利用 IP 核在 Mathlab 中快速完成数字信号处理的仿真和最终 FPGA 实现
Xilinx	Foundation	早期开发软件，目前已停止开发，转向 ISE
	ISE	新一代 FPGA/PLD 开发软件
	ISE Webpack	提供的免费软件，可从公司网站下载
	System Generator For DSP	配合 mathlab，在 FPGA 中完成数字信号处理的工具
Lattice	Isp Design EXPERT	早期的 PLD 开发软件
	Isp LEVER	取代 isp EXPERT，成为 FPGA 和 PLD 设计主要工具

Altera 开发的 Maxplus II 提供与结构无关的设计环境，能方便地进行设计输入、快速处理和器件编程。最新的 Quartus II 包括许多新的功能，将设计性能改善 15%，编译时间缩短 50%，除了支持 Altera 的 APEX 20KE、APEX 20KC、APEX II、ARM 的 Excalibur 嵌入处理器、Mercury、Stratix、FLEX10KE 和 ACEX1K 之外，还支持 MAX3000、MAX7000 系统列乘积项器件。

Xilinx 是 FPGA 的发明者，产品种类较全，主要有：XC9500，Coolrunner，Spartan，Virtex 等。其推出的 ISE 系列软件支持公司的所有 CPLD/FPGA 产品。通常来说，全球 PLD/FPGA 产品 60% 以上是由 Altera 和 Xilinx 提供的，可以说它们共同决定了 PLD 技术的发展方向。

Lattice 是 ISP（在系统编程）技术的发明者，ISP 技术极大地促进了 PLD 产品的发展，与 Altera 和 Xilinx 相比，其开发工具略逊一筹，中小规模 PLD 比较有特色。其他常见的 EDA 开发软件还有 orCAD/PSPIICE（最新版 10.5）、Multisim（最新版 9.0）、Protel（最新版 2006）等。

4. 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（类似于用于单片机开发的仿真器），供硬件验证用。一般包括：实验或开发所需的各类基本信号发生模块，如时钟、脉冲、高低电平；FPGA/CPLD 输出信息显示模块，如数据显示、发光管显示、声响指示等；监控程序模块，提供“电路重构软配置”；目标芯片适配座以及上面的芯片和编程下载电路。

1.3.3 EDA 设计流程

其设计流程图如图 1-3 所示。

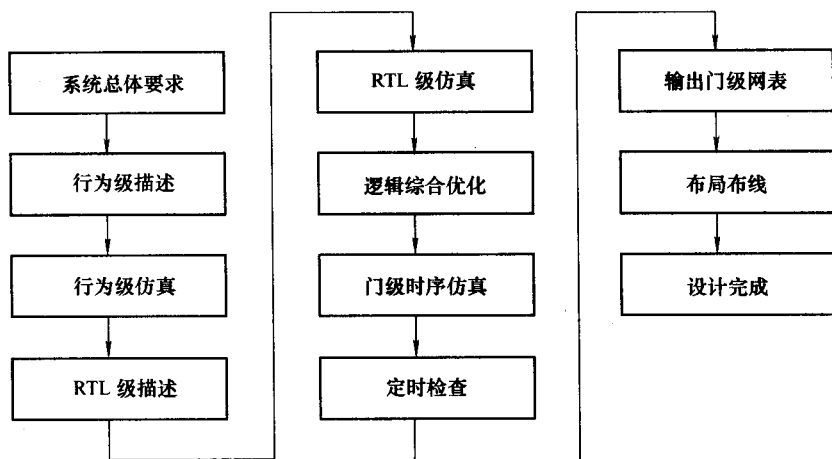


图 1-3 EDA 设计流程

首先从系统设计入手，在顶层进行功能框图的划分和结构设计，在框图一级进行仿真、纠错，并用硬件描述语言（HDL）对高层次的系统行为进行描述，在系统一级进行验证。然后用综合优化工具生成具体门电路的网表，其对应的物理实现可以是印制电路板或专用集成电路（ASIC）。由于设计的主要仿真和调试过程是在高层次上完成的，这不仅有利于早期

发现结构设计上的错误,避免设计工作中的浪费,而且也减少了逻辑功能仿真的工作量,提高了设计的一次性成功率。主要分为三个步骤:

1. 行为描述

目的是在系统设计的初始阶段,通过对行为描述的仿真来发现设计中存在的问题。考虑系统的结构及其工作过程是否能达到系统设计规格书的要求,并不考虑实际操作和算法用什么方法实现。它是对整个系统的数学模型的描述,与器件工艺无关。

2. 寄存器传输描述 RTL (又称数据流描述)

将行为方式描述的 HDL 程序,采用 RTL 方式描述,导出系统的逻辑表达式,再用仿真工具对 RTL 方式程序仿真。

3. 逻辑综合优化

利用逻辑综合工具,将 RTL 方式描述的程序转换成用基本逻辑元件表示的文件(门级网络表),也可以逻辑原理图方式输出。对逻辑综合结果在门电路级上仿真,并检查定时关系。

如果在某一层仿真发现问题,应返回上一层,寻找和修改相应错误,再向下继续。输出网表后,有两种选择:一种是由自动布线程序将网表转换成相应的 ASIC 芯片的制造工艺,定制 ASIC 芯片;另一种是将网表转换成相应的 PLD 编程码点,利用 PLD 完成硬件电路的设计。

1.4 EDA 应用与发展趋势

EDA 在教学、科研、产品设计与制造等方面都发挥着巨大的作用。几乎所有理工科(特别是电子信息)类的高校都开设了 EDA 课程,目的是培养学生使用 HDL 语言、EDA 工具设计简单系统的能力。从应用领域看,EDA 已渗透到各行各业,包括在机械、电子、通信、航空航天、化工、矿产、生物、医学、军事等。其发展趋势是政府重视、使用普及、应用广泛、工具多样、软件功能强大。

中国 EDA 市场已渐趋成熟,不过大部分设计工程师面向的是 PC 主板和小型 ASIC 领域,仅有小部分(约 11%)的设计人员开发复杂的片上系统器件。为了与台湾和美国的设计工程师形成更有力的竞争,中国的设计队伍有必要采用一些最新的 EDA 技术。

在 ASIC 和 PLD 芯片方面,向超高速、高密度、低功耗、低电压方向发展。在设计方面,继续向 SOPC 方向迈进,并且随着开发软件的发展,比如在软件设计中才用到的 C 语言开发工具,现在已经被用于硬件描述。硬件设计跟软件设计的界限将会变得越来越模糊。相信在不久的将来会有更多更好的设计工具。

第2章 VHDL 语言基础

硬件描述语言 VHDL (Very-High-Speed Integrated Circuit Hardware Description Language) 诞生于 1982 年。1987 年底, VHDL 被 IEEE 和美国国防部确认为标准硬件描述语言。自 IEEE 公布了 VHDL 的标准版本 IEEE—1076 (简称 87 版) 之后, 各 EDA 公司相继推出了自己的 VHDL 设计环境, 或宣布自己的设计工具可以和 VHDL 接口。此后 VHDL 在电子设计领域得到了广泛的认同, 并逐步取代了原有的非标准硬件描述语言。1993 年, IEEE 对 VHDL 进行了修订, 从更高的抽象层次和系统描述能力上扩展 VHDL 的内容, 公布了新版本的 VHDL, 即 IEEE 标准的 1076—1993 版本 (简称 93 版)。

2.1 硬件描述语言特点

VHDL 主要用于描述数字系统的结构、行为、功能和接口。VHDL 的语言形式、描述风格和句法与计算机高级程序语言非常类似, 不同的是, VHDL 语言中很多语句具有硬件特征。从执行方式上看, 一般的程序语言是顺序执行方式, 而 VHDL 语言是并行执行方式。应用 VHDL 语言进行数字系统设计, 具有以下突出优点:

- 1) 系统硬件描述能力强, 适于大型项目与团队合作开发。
- 2) 强大的行为描述能力可以避开具体的底层器件结构的设计。
- 3) 设计具有独立性, 设计者可以不懂硬件的结构, 也不必管理最终设计实现的目标器件是什么, 而进行独立的设计。
- 4) VHDL 语言符合 IEEE 工业标准, 编写的模块容易实现共享和复用。
- 5) 丰富的仿真语句和库函数, 使得任何大系统的设计在早期就能查验功能可行性, 随时可对设计进行仿真模拟。
- 6) 程序可读性好, 符合人类的思维习惯。

2.2 VHDL 程序基本结构

下面我们通过一个简单的二路选择器例子, 来说明一般 VHDL 程序的基本结构。

图 2-1 所示的二路选择器, 输入端为二个数据端口 d0、d1 和一个控制端口 sel, 输出端为 q。这个二路选择器要完成的工作可以描述为“q 输出端根据控制端口 sel, 选择相应的输入端数据进行输出”, 要搭建这样一个多路选择器模块, 在我们未接触 VHDL 语言前, 可以用与门、非门、或门等具体的电路底层器件按照图 2-2 的连接方式组成, 而硬件描述语言的出现, 可以使我们彻底摆脱具体的电路底层器件。

采用 VHDL 语言描述的二路选择器如下:

【例 2-1】 二路选择器的 VHDL 程序 (“--” 在 EDA 工具中, 表示注释)

```
LIBRARY IEEE;                                --库的调用
```

```

USE IEEE.std_logic_1164.ALL; --程序包的调用
ENTITY MUX2 IS                -- 实体说明
    PORT ( d0, d1:  IN  std_logic;
            sel:   IN  std_logic;
            q:    OUT std_logic);
END ENTITY;
ARCHITECTURE behav OF MUX2 IS --结构体描述
BEGIN
    PROCESS (d0, d1, sel)
    BEGIN
        IF sel = '0' THEN
            q <= d0;
        ELSIF sel = '1' THEN
            q <= d1;
        ELSE q <= 'Z';
        END IF;
    END PROCESS;
END behav;

```

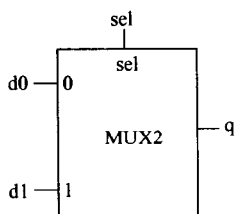


图 2-1 二路选择器模块

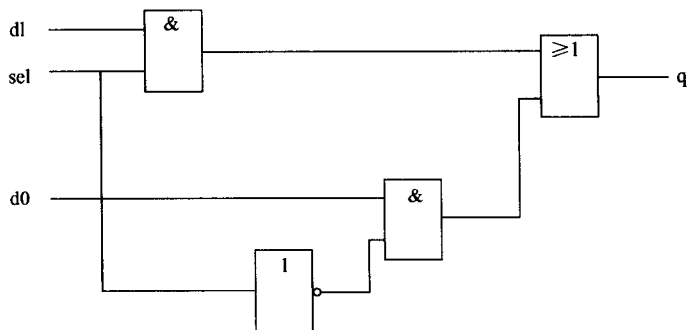


图 2-2 二路选择器底层硬件结构

用 VHDL 语言来描述的二路选择器不需要设计者具备底层的硬件知识，整个描述符合人的思维习惯。从这个例子我们可以总结得到，一个完整的 VHDL 程序包括库的调用、程序包的调用、实体说明和结构体描述 4 个部分。

库和库中程序包的调用类似于高级程序语言的文件头，程序中的函数及一些数据类型如 `std_logic` 等都在库中的程序包中有定义，因此程序要用到这些函数及数据类型则必须调用库和库的程序包。

实体是 VHDL 程序的基本单元，用于说明设计系统的外部接口信息，相当于提供一个设计单元的公共信息。对于一个已经确定的系统，实体的描述是唯一的。

结构体用于描述相应实体的行为、功能或电路结构，特别要注意的是结构体与实体不是一一对应的，一个实体可以对应多个结构体，但一个结构体只能对应一个实体。当一个实体