



21世纪普通高等教育规划教材

# EDA与

# 数字系统设计

李国丽 朱维勇 栾 铭 编著



机械工业出版社  
CHINA MACHINE PRESS

21 世纪普通高等教育规划教材

# EDA 与数字系统设计

李国丽 朱维勇 栾 铭 编著



机械工业出版社

本教材根据电子技术基础课程设计阶段学时少、任务重的特点，将传统电子技术课程设计与 EDA 技术有机结合，优化训练和设计内容，以提高将 EDA 技术用于数字系统设计的能力为目的，深入浅出地对 EDA 技术和相关知识做了介绍。

本教材介绍了 EDA 的相关知识、数字系统设计的基本概念和 MAX+plus2 的使用练习，介绍了 VHDL 和 Verilog HDL 硬件描述语言，并给出了若干数字系统设计问题。全书内容共 4 章，附录中给出了一些数字系统设计问题的 Verilog HDL 参考代码。

本书可以作为工科电类或非电类专业的本科生以及电大学生的电子技术课程设计教材或选修课教材，也可供有关教师和工程技术人员参考。

## 图书在版编目 (CIP) 数据

EDA 与数字系统设计/李国丽等编著. —北京: 机械工业出版社, 2004.1

(21 世纪普通高等教育规划教材)

ISBN 7-111-13071-5

I. E... II. 李... III. 硬件描述语言—数字电路—系统设计: 计算机辅助设计 IV. TN790.2

中国版本图书馆 CIP 数据核字 (2003) 第 083121 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: 周 娟

封面设计: 张 静 责任印制: 路 琳

北京蓝海印刷有限公司印刷 • 新华书店北京发行所发行

2004 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·15.25 印张·373 千字

定价: 23.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

# 前 言

随着电子技术的发展,特别是集成电路技术的发展,使电子电路逐步告别分立元件时代,向小型化、集成化方向发展,要开发拥有我国自主知识产权的硬件电路,就必须进行 ASIC、SOC 的设计,就必须使用和掌握 EDA 技术,这是科学技术的发展对我们的教学内容提出的新要求。为适应新世纪人才培养的需要,培养学生利用计算机等先进设备进行电子设计自动化的能力,合理安排教学和训练内容,使学生在最短的时间内,以最快的速度掌握 EDA 的基本方法,我们编写了这本教材,希望通过本教材的学习,可以使学生掌握 EDA 的一般方法,学习一个专用 EDA 软件 (MAX+plus2),并在有限的学时内完成数字系统设计课题。

VHDL 和 Verilog HDL 是目前 EDA 设计中常用的两种硬件描述语言,本教材对这两种硬件描述语言分别用一章的内容进行介绍,以供使用本教材的读者选择。

全书共分为 4 章。

绪论介绍数字系统设计的基本概念、数字系统设计方法简介、可编程逻辑器件简介、EDA 软件的种类及其各自的特点、硬件描述语言简介以及本教材教学的要求。

第 1 章通过练习进行 MAXPLUS2 的使用训练,练习是经过精心挑选的,只有完成这些练习,才能掌握 MAXPLUS2 软件的使用方法、下载箱的特点,才有可能进一步完成综合设计题。在此章的内容中,将出现一些简单的用 HDL 语言进行的设计,读者完全可以通过这些简单练习掌握一般问题的 HDL 设计。若需要对 HDL 语言有进一步的了解,可以参考第 2、3 章的有关内容。

第 2 章介绍 VHDL 硬件描述语言。以尽量简练的方法介绍 VHDL 语言的程序结构、基本数据类型和数据对象定义。通过大量的实例来说明 VHDL 语言的基本语法结构、基本语句和 VHDL 的程序设计特点等。并给出数字系统中常用电路的 VHDL 设计程序和用 MAX+PLUS 2 的仿真结果。

第 3 章介绍 Verilog HDL 语言,按照 IEEEStd1364: 1995 标准,介绍 Verilog HDL 的基础概念、语法要素、基本语句及仿真验证,并给出设计实例。

第 4 章给出数字系统设计若干问题,分别给出设计任务、设计提示和设计框图,要求学生根据前 3 章的内容在 EDA 软件 MAX+plus2 下,用不同的方法(图形输入法、HDL 硬件描述语言输入法)完成并下载实现它们。

附录给出了部分数字系统设计问题的 Verilog HDL 参考代码。

本教材的绪论、第 1 章、第 4 章及附录由合肥工业大学电气与自动化工程学院李国丽编写;第 2 章由朱维勇编写;第 3 章由栾铭编写,李伟协助完成;张柯、阎煜对书中的 Verilog HDL 代码的验证做了大量的工作。东南大学的黄正瑾教授审阅了本书的编写大纲并提出了宝贵意见,《电气电子教学学报》孙文治主编对这本书的编写给予了很大的关心,解放军电

#### IV

子工程学院的李东生教授审阅了书稿，并提出了宝贵意见。在本书的编写过程中，参考和引用了有关专家的教材和论文，在此一并表示衷心的感谢。

由于编者的水平有限，书中可能存在不妥之处，欢迎读者批评指正。

编者

# 目 录

## 前言

绪论 ..... 1

0.1 数字系统设计的基本概念 ..... 1

0.2 数字系统设计方法简介 ..... 2

0.3 可编程逻辑器件简介 ..... 4

0.4 EDA 软件种类及其各自特点 ..... 5

0.5 硬件描述语言简介 ..... 7

第 1 章 MAX+plus2 使用练习 ..... 9

1.1 MAX+plus2 简介 ..... 9

1.2 基于 MAX+plus2 的电路设计过程 ..... 10

1.3 计数器设计 ..... 21

1.4 扫描显示电路 ..... 24

1.5 数字系统设计例题 ..... 31

第 2 章 VHDL 硬件描述语言 ..... 37

2.1 VHDL 概述 ..... 37

2.1.1 VHDL 的特点 ..... 37

2.1.2 VHDL 语言的基本结构 ..... 37

2.1.3 VHDL 语言的实体  
(ENTITY) 说明语句 ..... 39

2.1.4 VHDL 语言的结构体  
(ARCHITECTURE) ..... 40

2.1.5 程序包 (PACKAGE)、库  
(LIBRARY) 和 USE 语句 ..... 41

2.2 VHDL 的数据类型  
和数据对象 ..... 43

2.2.1 VHDL 的标记 ..... 43

2.2.2 VHDL 的数据类型 ..... 44

2.2.3 VHDL 的运算符 ..... 49

2.2.4 VHDL 的数据对象 ..... 53

2.3 VHD 设计的基本语句 ..... 57

2.3.1 并行信号赋值语句 ..... 57

2.3.2 条件赋值语句 WHEN

-ELSE ..... 58

2.3.3 选择信号赋值语句

WITH-SELECT ..... 59

2.3.4 块 (BLOCK) 语句 ..... 59

2.3.5 IF-ELSE 语句 ..... 61

2.3.6 CASE-WHEN 语句 ..... 63

2.3.7 FOR-LOOP 语句 ..... 64

2.4 VHDL 高级语句 ..... 65

2.4.1 进程 (PROCESS) 语句 ..... 65

2.4.2 元件 (COMPONENT) 定义  
语句和元件例化 (PORT MAP)  
语句 ..... 66

2.4.3 生成 (GENERATE) 语句 ..... 68

2.4.4 子程序 (SUBPROGRAM) ..... 71

2.4.5 程序包的设计 ..... 75

2.5 VHDL 设计实例 ..... 76

2.5.1 常见的组合逻辑电路设计 ..... 76

2.5.2 常见的时序逻辑电路设计 ..... 83

2.5.3 状态机设计 ..... 93

第 3 章 Verilog HDL 硬件描述语言 ..... 98

3.1 Verilog HDL 概述 ..... 98

3.1.1 Verilog HDL 的特点 ..... 98

3.1.2 Verilog HDL 的基本结构 ..... 98

3.2 Verilog HDL 语言要素 ..... 99

3.2.1 基本语法定义 ..... 99

3.2.2 数据类型 ..... 104

3.2.3 系统任务与系统函数 ..... 106

3.2.4 编译向导 ..... 110

3.3 Verilog HDL 基本语句 ..... 112

3.3.1 过程语句 (Structured Procedure) .....	112	4.7 脉冲按键电话按键显示器 .....	169
3.3.2 赋值语句 .....	113	4.7.1 设计要求 .....	169
3.3.3 块语句 (Block Statements) ...	117	4.7.2 设计提示 .....	169
3.3.4 条件语句 (Conditional Statement) .....	119	4.8 乘法器 .....	170
3.3.5 循环语句 .....	123	4.8.1 设计要求 .....	170
3.3.6 任务与函数 .....	125	4.8.2 设计提示 .....	171
3.4 仿真验证 (Simulation) .....	128	4.9 乒乓球比赛游戏机 .....	171
3.5 可综合性描述 (Coding for Synthesis) .....	138	4.9.1 设计要求 .....	171
3.6 设计实例 .....	146	4.9.2 设计提示 .....	172
3.6.1 译码电路 .....	146	4.10 具有四种信号灯的交通灯 控制器 .....	174
3.6.2 编码电路 .....	147	4.10.1 设计要求 .....	174
3.6.3 数据分配器 .....	147	4.10.2 设计提示 .....	174
3.6.4 同步计数器 .....	149	4.11 出租车自动计费器 .....	175
3.6.5 移位寄存器 .....	151	4.11.1 设计要求 .....	175
3.6.6 有限状态机的设计 .....	152	4.11.2 设计提示 .....	175
3.6.7 复杂逻辑电路的设计 .....	155	4.12 自动售邮票机 .....	177
<b>第4章 数字系统设计问题</b> .....	<b>158</b>	4.12.1 设计要求 .....	177
4.1 多功能数字钟的设计 .....	158	4.12.2 设计提示 .....	177
4.1.1 设计要求 .....	158	4.13 电梯控制器 .....	178
4.1.2 设计提示 .....	158	4.13.1 设计要求 .....	178
4.2 数字式竞赛抢答器 .....	160	4.13.2 设计提示 .....	179
4.2.1 设计要求 .....	160	<b>附录</b> .....	<b>181</b>
4.2.2 设计提示 .....	160	附录 A 多功能数字钟主控电路 Verilog HDL 参考代码 .....	181
4.3 数字频率表 .....	161	附录 B 数字式竞赛抢答器主控电路 Verilog HDL 参考代码 .....	186
4.3.1 设计要求 .....	161	附录 C 洗衣机控制器主控电路 Verilog HDL 参考代码 .....	192
4.3.2 设计提示 .....	161	附录 D 电子密码锁主控电路 Verilog HDL 参考代码 .....	196
4.4 拔河游戏机 .....	163	附录 E 乘法器主控电路 Verilog HDL 参考代码 .....	198
4.4.1 设计要求 .....	163	附录 F 乒乓球比赛游戏机主控电路 Verilog HDL 参考代码 .....	202
4.4.2 设计提示 .....	164	附录 G 具有四种信号灯的交通灯 控制器主控电路 Verilog HDL 参考代码 .....	205
4.5 洗衣机控制器 .....	164		
4.5.1 设计要求 .....	164		
4.5.2 设计提示 .....	165		
4.6 电子密码锁 .....	167		
4.6.1 设计要求 .....	167		
4.6.2 设计提示 .....	168		

附录 H 出租车自动计费器主控电路		附录 J 电梯控制器主控电路 Verilog	
Verilog HDL 参考代码 .....	208	HDL 参考代码 .....	214
附录 I 自动售邮票机主控电路 Verilog		附录 K VHDL 标准程序包文件 .....	218
HDL 参考代码 .....	212	参考文献 .....	234



# 绪 论

## 0.1 数字系统设计的基本概念

目前，数字技术已渗透到科研、生产和人们日常生活的各个领域。从计算机到家用电器，从手机到数字电话，以及绝大部分新研制的医用设备、军用设备等，无不尽可能地采用了数字技术。

数字系统是对数字信息进行存储、传输、处理的电子系统。

通常把门电路、触发器等称为逻辑器件。将由逻辑器件构成，能执行某单一功能的电路，如计数器、译码器、加法器等，称为逻辑功能部件，把由逻辑功能部件组成的能实现复杂功能的数字电路称为数字系统。复杂的数字系统可以分割成若干个子系统，例如计算机就是一个内部结构相当复杂的数字系统。

不论数字系统的复杂程度如何，规模大小怎样，就其实质而言皆为逻辑问题，从组成上说，是由许多能够进行各种逻辑操作的功能部件组成的，这类功能部件，可以是小规模集成电路（SSI）逻辑部件，也可以是各种 MSI、LSI 逻辑部件，甚至可以是 CPU 芯片。由于各功能部件之间的有机配合，协调工作，使数字电路成为统一的数字信息存储、传输、处理的电子电路。

与数字系统相对应的是模拟系统，和模拟系统相比，数字系统具有工作稳定可靠，抗干扰能力强，便于大规模集成，易于实现小型化、模块化等优点。

数字系统一般由控制电路、多个受控电路、输入/输出电路、时基电路等几部分构成，如图 0-1 所示。

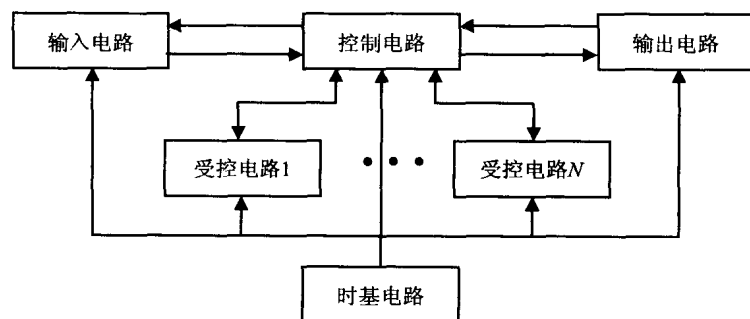


图 0-1 数字系统框图

图中，输入电路将外部信号（开关信号、时钟信号等）引入数字系统，经控制电路逻辑处理后，或控制受控电路，或经输出电路产生外部执行机构（发光二极管、数码管、扬声器等）所需的信号。数字系统通常是一个时序电路，时基电路产生各种时钟信号，保证

整个系统在时钟作用下协调工作。

数字系统和逻辑功能部件之间的区别之一在于功能是否单一，一个存储器，尽管规模很大，可以达到数兆甚至 G 字节，但因其功能单一，只能算是逻辑功能部件，而由几片 MSI 构成的交通灯控制器却应称为系统。

数字系统和功能部件之间的区别之二是是否包含控制电路，一个数字电路，无论其规模大小，只有在具有控制电路的情况下才能称之为系统。控制电路根据外部输入信号、各受控电路的反馈信号、控制电路的当前状态，决定系统的下一步动作。控制电路的逻辑关系最为复杂，是数字系统设计中的关键。

## 0.2 数字系统设计方法简介

数字系统设计的一般流程为

### 1. 明确设计要求，确定系统的输入/输出

在具体设计之前，详细分析设计要求，确定系统输入/输出信号是必要的。例如，要设计一个交通灯控制器，必须明确系统的输入信号有哪些（由传感器得到的车辆到来信号、时钟信号），输出要求是什么（红、黄、绿交通灯正确显示和时间显示），只有在明确设计要求的基础上，才能使系统设计有序地进行。

### 2. 确定整体设计方案

对于一个具体的设计，可能有多种不同的方案，确定方案时，应对不同方案的性能、成本、可靠性等方面进行综合考虑，最终确定设计方案。

### 3. 自顶向下（Top-down）的模块化设计方法

数字系统的设计通常有两种设计方法：一种是自底向上的设计方法；一种是自顶向下的设计方法。

自底向上（Bottom-up）的设计过程从最底层设计开始。设计系统硬件时，首先选择具体的元器件，用这些元器件通过逻辑电路设计，完成系统中各独立功能模块的设计，再把这些功能模块连接起来，总装成完整的硬件系统。

这种设计过程在进行传统的手工电路设计时经常用到，优点是符合硬件设计工程师传统的设计习惯；缺点是在进行底层设计时，缺乏对整个电子系统总体性能的把握，在整个系统设计完成后，如果发现性能尚待改进，修改起来比较困难，因而设计周期长。

随着集成电路设计规模的不断扩大，复杂度的不断提高，传统的电路原理图输入法已经无法满足设计的要求。EDA 工具和 HDL 的产生使自顶向下（Top-down）的设计方法得以实现。

自顶向下（Top-down）的设计方法是在顶层设计中，把整个系统看成是包含输入输出端口的单个模块，对系统级进行仿真、纠错，然后对顶层进行功能框图和结构的划分，即从整个系统的功能出发，按一定原则，将系统分成若干个子系统，再将每个子系统分成若干个功能模块，进而将每个模块分成若干个小的模块……直至分成许多基本模块实现。这样将系统模块划分为各个子功能模块，并对其进行行为描述，在行为级进行验证。

例如，交通灯控制器的设计，可以把整个系统分为主控电路、定时电路、译码驱动显

示等。定时电路可以由计数器功能模块构成，译码驱动显示可由 SSI 构成的组合逻辑电路组成，这两部分都是设计者所熟悉的各种功能电路，设计起来并不困难，这样交通灯控制器的设计的主要问题就是控制电路的设计了，而这是一个规模不大的时序电路，也就是说可以把一个复杂的数字系统的设计变成一个较小规模的时序电路的设计，从而大大简化设计的难度，缩短设计周期。由于设计调试都可以针对这些子模块进行，使修改设计也变得非常方便。

模块分割的一般要求为：

- 1) 各模块之间的逻辑关系明确。
- 2) 各模块内部逻辑功能集中，且易于实现。
- 3) 各模块之间的接口线尽量少。

模块化的设计最能体现设计者的思想，分割合适与否对系统设计的方便与否有着至关重要的影响，

#### 4. 数字系统的设计

数字系统的设计可以在以下几个层次上进行：

- 1) 选用通用集成电路芯片构成数字系统。
- 2) 应用可编程逻辑器件实现数字系统。
- 3) 设计专用集成电路（单片系统）。

用通用集成电路构成数字系统即采用 SSI、MSI、LSI（如 74 系列芯片，计数器芯片、存储器芯片等），根据系统的设计要求，构成所需数字系统。早期的数字系统的设计，都是在这个层次上进行的，电子工程师设计电子系统的过程一般是：根据设计要求，进行书面设计——选择器件——电路搭建调试——样机制作，这样完成的系统设计，由于芯片之间的众多连接，造成系统的可靠性不高，也使系统体积相对较大，集成度低。当数字系统大到一定规模时，搭建调试会变得非常困难甚至不可行。

随着数字集成技术和电子设计自动化（Electronic Design Automation, EDA）技术的迅速发展，数字系统设计的理论和方法也在相应地变化和发展着。EDA 技术是从计算机辅助设计（CAD）、计算机辅助制造（CAM）、计算机辅助测试（CAT）和计算机辅助工程（CAE）等技术发展而来的。它以计算机为工具，设计者只需对系统功能进行描述，就可在 EDA 工具的帮助下完成系统设计。

应用可编程逻辑器件（Programmable Logic Device, PLD）实现数字系统设计和单片系统的设计，是目前利用 EDA 技术设计数字系统的潮流。这种设计方法以数字系统设计软件为工具，将传统数字系统设计中的搭建调试用软件仿真取代，对计算机上建立的系统模型，用测试码或测试序列测试验证后，将系统实现在 PLD 芯片或专用集成电路上，这样最大程度地缩短了设计和开发时间，降低了成本，提高了系统的可靠性。

高速发展的可编程逻辑器件为 EDA 技术的不断进步奠定了坚实的物理基础。大规模可编程逻辑器件不但具有微处理器和单片机的特点，而且随着微电子技术和半导体制造工艺的进步，集成度不断提高，与微处理器、DSP、A/D、D/A、RAM 和 ROM 等独立器件之间的物理与功能界限正日趋模糊，嵌入式系统和片上系统（SOC）得以实现。以大规模可编程集成电路为物质基础的 EDA 技术打破了软硬件之间的设计界限，使硬件系统软件化。这已成为现代电子设计技术的发展趋势。

### 0.3 可编程逻辑器件简介

数字集成电路从它的产生到现在,经过了早期的电子管、晶体管、小中规模集成电路,到大规模、超大规模集成电路(VLSIC,几万门以上)以及许多具有特定功能的专用集成电路的发展过程。但是,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。系统设计师们更愿意自己设计专用集成电路(Application Specific Integrated Circuit, 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CPLD和FPGA。

CPLD是复杂可编程逻辑器件(Complex Programmable Logic Device)的简称,FPGA是现场可编程门阵列(Field Programmable Gate Array)的简称。两者的功能基本相同,只是实现原理略有不同,所以我们有时可以忽略这两者的区别,统称为可编程逻辑器件或CPLD/FPGA。

可编程逻辑器件是电子设计领域中最具活力和发展前途的一项技术,它的影响丝毫不亚于20世纪70年代单片机的发明和使用。

可编程逻辑器件能完成任何数字器件的功能,上至高性能CPU,下至简单的74电路,都可以用可编程逻辑器件来实现。可编程逻辑器件如同一张白纸或是一堆积木,工程师可以通过传统的原理图输入法或硬件描述语言,自由地设计一个数字系统,通过软件仿真,我们可以事先验证设计的正确性,还可以利用PLD的在线修改能力,随时修改设计。

使用可编程逻辑器件来开发数字电路,可以大大缩短设计时间,减少芯片面积,提高系统的可靠性。可编程逻辑器件的这些优点使得可编程逻辑器件技术在20世纪90年代以后得到飞速的发展,同时也大大推动了EDA软件和硬件描述语言(Hardware Description Language, HDL)的进步。

早期的可编程逻辑器件只有可编程只读存储器(PROM)、紫外线可擦除只读存储器(EPROM)和电可擦除只读存储器(EEPROM)三种,它们由全译码的与阵列和可编程的或阵列组成,由于阵列规模大,速度低,主要用途是作存储器。

20世纪70年代中期,出现了一类结构上稍复杂的可编程芯片,称为可编程逻辑阵列(Programmable Logic Array, PLA)。它由可编程的与阵列和可编程的或阵列组成,虽然其阵列规模大为减小,提高了芯片的利用率,但由于编程复杂,支持PLA的开发软件有一定难度,因而也没有得到广泛应用。

20世纪70年代末,美国MMI公司(Monolithic Memories Inc, 单片机存储器公司)率先推出了可编程阵列逻辑器件(Programmable Array Logic, PAL)。PAL由可编程的与阵列和固定的或阵列构成,采用熔丝编程方式、双极型工艺制造。PAL在器件的工作速度、输出结构种类上较早期的可编程逻辑器件有了很大进步,但由于其输出方式的固定不能重新组态,所以编程灵活性较差,又由于采用的是PROM工艺,只能一次性编程,使用者仍要承担一定风险。

20世纪80年代中期,Altera公司发明了通用阵列逻辑(Generic Array Logic, GAL)。它和PAL的区别在于GAL的输出电路可以组态,而且它大多采用UVC MOS或E<sup>2</sup>CMOS

工艺, 实现了重复编程, 通常可擦写百次以上, 甚至上千次。GAL 由于其设计具有很强的灵活性, 设计风险为零, 可以取代大部分 SSI、MSI、和 PAL 器件, 所以在 20 世纪 80 年代得到广泛使用。

这些早期的可编程逻辑器件的一个共同特点是都属于低密度 PLD, 结构简单, 设计灵活, 但规模小, 难以实现复杂的逻辑功能。

其后, 随着集成电路工艺水平的不断提高, PLD 突破了传统的单一结构, 向着高密度、高速度、低功耗以及结构体系更灵活、适用范围更宽的方向发展, 因而相继出现了各种不同结构的高密度 PLD。20 世纪 80 年代中期, Altera 公司推出了一种新型的可擦除、可编程逻辑器件 (Erasable Programmable Logic Device, EPLD), 它采用 CMOS 和 UVEPROM 工艺制作, 集成度比 PAL 和 GAL 高得多, 设计也更加灵活, 但内部互联能力比较弱。

1985 年 Xilinx 公司首家推出了现场可编程门阵列器件 (Field Programmable Gate Array FPGA), 它是一种新型的高密度 PLD, 采用 CMOS-SRAM 工艺制作, 其结构和阵列型 PLD 不同, 内部由许多独立的可编程逻辑模块组成, 逻辑模块之间可以灵活地相互连接, 具有密度高、编程速度快、设计灵活和可再配置设计能力等许多优点。FPGA 出现后立即受到世界范围内电子工程师的普遍欢迎, 并得到迅速发展。

20 世纪 80 年代末, Lattice 公司提出在系统可编程 (In System Programmable, ISP) 技术后, 相继出现了一系列具备在系统可编程能力的复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD), CPLD 是在 EPLD 的基础上发展起来的, 它采用  $E^2$ CMOS 工艺制作, 增加了内部连线, 改进了内部结构体系, 因而比 EPLD 性能更好, 设计更加灵活, 其发展也非常迅速。

不同厂家对可编程逻辑器件的叫法也不尽相同。Xilinx 公司把基于查找表技术, SRAM 工艺, 要外挂配置用的 EEPROM 的可编程逻辑器件称为 FPGA; 把基于乘积项技术, Flash 工艺 (类似 EEPROM 工艺) 的可编程逻辑器件称为 CPLD。Altera 公司把自己的可编程逻辑器件产品中的 MAX 系列 (乘积项技术, EEPROM 工艺)、FLEX 系列 (查找表技术, SRAM 工艺) 都叫做 CPLD; 而把也是 SRAM 工艺、基于查找表技术、要外挂配置用的 FLEX 系列的 EPROM 叫做 FPGA。

20 世纪 90 年代以后, 高密度 PLD 在生产工艺、器件的编程和测试技术等方面都有了飞速发展, CPLD 的集成度一般可达数千甚至上万门。Altera 公司的 EPM9560, 其单密度达到 12000 个可用门, 包含多达 50 个宏单元, 216 个用户 I/O 引脚, 并能提供 15ns 的脚至脚延时, 16 位计数的最高工作频率为 118MHz。目前, CPLD 的集成度最多可达 25 万个等效门以上, 最高工作速度已超过 180MHz, FPGA 的门延时已小于 3ns。在系统可编程技术、边界扫描技术的出现, 也使器件在编程技术和测试技术及系统可重构技术方面有了很快的发展。

## 0.4 EDA 软件种类及其各自特点

计算机技术的进步推动了 EDA 技术的普及和发展, EDA 工具层出不穷, 目前在我国各大大专院校教学中, 具有的广泛影响的 EDA 软件有 Pspice、OrCad、Electrical Workbench、Protel 等。

Pspice 是美国 MicroSim 公司于 20 世纪 80 年代开发的电路模拟分析软件, 可以进行模拟分析、模拟数字混合分析、参数优化等。该公司还开发了 PCB、CPLD 的设计软件, 该软件现已并入 OrCad。

OrCad 是一个大型的电子线路 EDA 软件包, OrCad 公司的产品包括原理图设计、PCB 设计、PLD Tools 等设计软件工具。OrCad 被 Cadence 公司收购后, 其产品功能更加强大。

Electronics Workbench 软件是加拿大 Interactive Image Technologies 公司于 20 世纪 80 年代末、90 年代初推出的专门用于电子线路仿真的“虚拟电子工作台”软件, 可以将不同类型的电路组合成混合电路进行仿真。它不仅可以完成电路的瞬态分析和稳态分析、时域和频域分析、器件的线性和非线性分析、电路的噪声分析和失真分析等常规电路的分析, 而且还提供了离散傅里叶分析、电路零极点分析、交直流灵敏度分析和电路容差分析等共计 14 种电路分析方法, 并具有故障模拟和数据储存等功能。其升级版本 Multisim 2001 除具备上述功能外, 还支持 VHDL 和 Verilog HDL 文本的输入。

Protel 软件包是 20 世纪 90 年代初由澳大利亚 Protel Technology 公司研制开发的电路 EDA 软件, 它在我国电子行业中知名度很高, 普及程度较广。Protel 98 是应用于 Windows 95 / 98 和 Windows NT 下的 EDA 设计软件, 它包括 5 大组件: 原理图设计系统 Advanced Schematic 98、印制电路板设计系统 Advanced PCB 98、可编程逻辑器件(PLD)设计系统 Advanced PLD 98、电路仿真系统 Advanced SIM 98 以及自动布线系统 Advanced Route 98。它可以完成电路原理图的设计和绘制、电路仿真、印制电路板设计、可编程逻辑器件(PLD)设计和自动布线等。在 Protel 98 的基础上, Protel 经历了 Protel 99、Protel 99se、Protel DXP 的发展过程, 功能也愈来愈完善。

除此之外, 专门用于开发 FPGA 和 CPLD 的 EDA 工具也很多, 它们大致可以分为五个模块:

- 1) 设计输入编辑器;
- 2) 仿真器;
- 3) HDL 综合器;
- 4) 适配器 (或布局布线器);
- 5) 下载器。

设计输入编辑器可以接受不同的设计输入方式, 如原理图输入方式、状态图输入方式、波形图输入方式以及 HDL 文本输入方式。各 PLD 厂商一般都有自己的设计输入编辑器, 如 Xilinx 公司的 Foundation、Altera 公司的 MAX+plus2 等。

基于 HDL 的仿真器应用广泛。数字系统的设计中, 行为模型的表达、电子系统的建模、逻辑电路的验证以及门级系统的测试, 都离不开仿真器的模拟检测。按处理的硬件描述语言, 仿真器可分为 VHDL 仿真器、Verilog 仿真器等; 按仿真的电路描述级别的不同, HDL 仿真器可以独立或综合完成系统级仿真、行为级仿真、RTL 级仿真和门级时序仿真。

各 EDA 厂商都提供基于 VHDL/Verilog 的仿真器, 如 Mentor 公司的 ModelSim, Cadence 公司的 Verilog-XL、NC-Verilog, Synopsys 公司的 VCS, Aldec 公司的 Active-HDL 等。

HDL 综合器可以把 VHDL/Verilog HDL 描述的系统转化成硬件电路, 这样使硬件描述语言不仅适用于电路逻辑的建模和仿真, 还可以直接用于电路的设计。目前常用的

FPGA/CPLD 设计的 HDL 综合器为:

- 1) Synopsys 公司的 FPGA Compiler、FPGA Express。
- 2) Synplicity 公司的 Synplify Pro 综合器。
- 3) Mentor 子公司 Exemplar Logic 的 Leonardo Spectrum 综合器。

综合器综合电路时,首先对 VHDL/Verilog 进行分析处理,并将其转换成相应的电路结构或模块,这是一个通用电路原理图的形成过程,与硬件无关。然后才对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。

综合器一般输出网表文件,如 EDIF (Electronic Design Interchange Format)。文件后缀是.edf,或是直接用 VHDL/Verilog 语言表达的标准格式的网表文件,或是对应 FPGA 器件厂商的网表文件,如 Xilinx 公司的 XNF 网表文件。

适配器又称为布局布线器,其任务是完成系统在器件上的布局布线。适配器输出的是厂商自己定义的下载文件,用于下载到器件中,以实现设计。布局布线通常由 PLD 厂商提供的专门针对器件开发的软件来完成,这些软件可以嵌在 EDA 开发环境中,也可以是专用的适配器。例如,Lattice 公司的 ispEXPERT、Altera 公司的 MAX+plus2 和 Quartus、Xilinx 公司的 Foundation 和 ISE 中都有各自的适配器。

下载器又称为编程器,它把设计下载到对应的实际器件中,实现硬件设计,一般 PLD 厂商都提供专门针对器件的下载或编程软件。

## 0.5 硬件描述语言简介

数字系统的设计输入方式有多种,通常是由线信号和表示基本设计单元的符号连在一起组成线路图,符号取自器件库,符号通过信号(或网线)连接在一起,信号使符号互连,这样设计的系统所形成的设计文件是若干张电路原理结构图,在图中详细标注了各逻辑单元、器件的名称和相互间的信号连接关系。对于小的系统,这种原理电路图只要几十张至几百张就可以了,但如果系统比较大,硬件比较复杂,这样的原理电路图可能要几千张、几万张甚至更多,这样就给设计归档、阅读、修改等都带来了不便。这一点在 IC 设计领域表现得尤为突出,从而导致了采用硬件描述语言进行硬件电路设计方法的兴起。

硬件描述语言(Hardware Description Language, HDL)是用文本形式来描述数字电路的内部结构和信号连接关系的一类语言,类似于一般的计算机高级语言的语言形式和结构形式。设计者可以利用 HDL 描述设计的电路,然后利用 EDA 工具进行综合和仿真,最后形成目标文件,再用 ASIC 或 PLD 等器件实现。

硬件描述语言的发展至今约有 20 多年的历史,并成功地应用于数字系统开发的各个阶段:设计、综合、仿真和验证等,使设计过程达到高度自动化。硬件描述语言有多种类型,最具代表性的、使用最广泛的是 VHDL (Very High Speed Intergated Circuit Hardware Description Language)语言和 Verilog HDL 语言。

VHDL 语言于 20 世纪 80 年代初由美国国防部(The United States Department of Defense)发起创建,当时制定了一个名为 VHSIC (Very High Speed Integrated Circuit)的计划,其目的是为了能制定一个标准的文件格式和语法,要求各武器承包商遵循该标准描述其设计的电路,以便于保存和重复使用电子电路设计。VHDL 语言的全称为“超高速集

成电路硬件描述语言”(VHSIC Hardware Description Language), 于 1982 年正式诞生, VHDL 吸取了计算机高级语言语法严谨的优点, 采用了模块化的设计方法, 于 1987 年被国际电气电子工程师学会 (International Electrical & Electronic Engineering, IEEE) 收纳为标准; 文件编号为 IEEE Standard 1076。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了 VHDL 的内容, 公布了新版本的 VHDL, 即 IEEE 标准的 1076-1993 版本。

Verilog HDL 语言最初是于 1983 年由 Gateway Design Automation (GDA) 公司的 Phil Moorby 为其模拟器产品开发的硬件描述语言, 那时它只是一种专用语言, 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 推出它的第三个商用仿真器 Verilog-XL, 获得了巨大的成功, 由于他们的模拟、仿真器产品的广泛使用, Verilog HDL 作为一种便于实用的语言逐步为设计者所接受。1989 年 Cadence 公司收购了 GDA 公司, 使得 Verilog HDL 成为该公司的专有技术。1990 年 Cadence 公司公开发表了 Verilog HDL, 并成立 OVI(Open Verilon International)来促进 Verilog HDL 的发展, 致力于推广 Verilog HDL 成为 IEEE 标准, 这一努力最后获得成功, Verilog 语言于 1995 年成为 IEEE 标准, 称为 IEEE Std 1364-1995。

本书将在介绍数字系统的设计方法及基本步骤的基础上, 介绍 MAX+plus2 的使用方法, 介绍硬件描述语言 VHDL 和 Verilog HDL, 并给出若干数字系统设计问题, 期望通过实例和练习, 把数字系统设计的基本理论、基本方法和设计课题紧密结合, 使读者在 MAX+plus2 的设计平台下, 学会用原理电路图输入或硬件描述语言输入 (VHDL 或 Verilog HDL) 进行电路设计、编译 (Compiler)、仿真 ((Simulator)、底层编辑 (Floorplan Editor) 及 PLD 编程校验 (Programmer 或 Configure), 对波形编辑器 (Waveform Editor)、逻辑综合与试配 (Logic Synthesize) 等涉及不多, 以求提高读者利用 MAX+plus2 进行数字系统设计的能力。

本书除第 3 章使用 Mentor 公司的 Modelsim 仿真器、Synplicity 公司的 Synplify 综合器进行仿真和综合外, 其余章节的仿真结果都是用 Altera 公司的 MAX+plus2 得到的。



# 第 1 章 MAX+plus2 使用练习

## 1.1 MAX+plus2 简介

MAX+plus2 (Multiple Array Matrix and programmable logic user system) 是一个完全集成化、易学易用的可编程逻辑设计环境, 它可以在多种平台上运行, 其图形界面丰富, 加上完整的、可即时访问的在线文档, 使设计人员可以轻松地掌握软件的使用。

MAX+plus2 开发系统有很多特点:

### 1. 界面开放

MAX+plus2 是 Altera 公司的 EDA 软件, 但它可以与其他工业标准的设计输入、综合与校验工具相连接, 设计人员可以使用 Altera 或标准 EDA 设计输入工具来建立逻辑设计, 用 MAX+plus2 编译器 (Compiler) 对 Altera 器件设计进行编译, 并使用 Altera 或其他 EDA 校验工具进行器件或板级仿真。目前, MAX+plus2 支持与 Cadence、Exemplarlogic、Mentor Graphics、Synopsys、Synplicity、Viewlogic 等公司所提供的 EDA 工具接口。

### 2. 与结构无关

MAX+plus2 系统的核心编码器 (Compiler) 支持 Altera 公司的 FLEX10K、FLEX8000、FLEX6000、MAX9000、MAX7000、MAX5000 和 Classic 可编程逻辑器件系列, 提供了与结构无关的可编程逻辑环境。MAX+plus2 的编译器还提供了强大的逻辑综合与优化功能, 使用户可以容易地把设计集成到器件中。

### 3. 丰富的设计库

MAX+plus2 提供丰富的库元器件供设计者调用, 其中包括 74 系列的全部器件和其他多种逻辑功能部件, 调用库元器件进行设计, 可以大大减轻设计人员的工作量, 缩短设计周期。

### 4. 模块化工具

设计人员可以从各种设计输入、处理和校验选项中进行选择, 从而使 MAX+plus2 可以满足不同用户的需求, 根据需要, 还可以添加新功能。例如, 在本教材中, 侧重点在于用 MAX+plus2 进行各种设计输入 (图形或 HDL 输入)、编译 (Compiler)、仿真 (Simulator)、底层编辑 (Floorplan Editor) 及 PLD 器件编程校验 (Programmer 或 Configure), 并不过多涉及波形编辑器 (Waveform Editor)、逻辑综合 (Logic Synthesize) 与试配等。

### 5. 硬件描述语言

MAX+plus2 软件支持各种 HDL 设计输入选项, 包括 VHDL、Verilog HDL 和 Altera 公司的 AHDL。

MAX+plus2 软件的启动开始界面如图 1-1-1 所示。