



“十二五”普通高等教育本科国家级规划教材

EDA与 数字系统设计

第③版

李国丽 朱维勇 编著

EDA and Digital System Design



 机械工业出版社
CHINA MACHINE PRESS

“十二五”普通高等教育本科国家级规划教材

EDA 与数字系统设计

第3版

李国丽 朱维勇 编著



机械工业出版社

本书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,以提高学生将 EDA 技术用于数字系统设计的能力为目的,深入浅出地介绍 EDA 技术和相关知识。

本书共分为 4 章,主要内容包括 EDA 的相关知识、数字系统设计的基本概念和 Quartus II 的使用练习, VHDL 和 Verilog HDL,并给出若干数字系统设计题目。附录中给出了一些数字系统设计实例的 VHDL 和 Verilog HDL 参考代码。

本书可作为工科电类或非电类专业本科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。

本书配有免费电子课件,欢迎选用本书作教材的教师登录 www.cmpedu.com 注册下载。

图书在版编目 (CIP) 数据

EDA 与数字系统设计/李国丽,朱维勇编著. —3 版. —北京:机械工业出版社,2019.3

“十二五”普通高等教育本科国家级规划教材

ISBN 978-7-111-61986-4

I. ①E… II. ①李… ②朱… III. ①电子电路-计算机辅助设计-应用软件-高等学校-教材 ②数字系统-系统设计-高等学校-教材
IV. ①TN702.2②TP271

中国版本图书馆 CIP 数据核字 (2019) 第 025861 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:王玉鑫 责任编辑:王玉鑫

责任校对:郑婕 封面设计:张静

责任印制:张博

唐山三艺印务有限公司印刷

2019 年 5 月第 3 版第 1 次印刷

184mm×260mm·19.75 印张·484 千字

标准书号:ISBN 978-7-111-61986-4

定价:49.80 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

网络服务

服务咨询热线:010-88379833

机工官网:www.cmpbook.com

读者购书热线:010-68326294

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面防伪标均为盗版

金书网:www.golden-book.com

前 言

随着计算机技术、集成电路技术的飞速发展，新器件不断出现，所有 EDA 工具软件都不断地更新换代。为适应新世纪人才培养的需要，提高学生电子设计自动化的能力，使学生在最短的时间内，以最快的速度掌握基于 Quartus II 的 EDA 技术，我们对本书的第 2 版进行了再次修订。

书中除了介绍 Quartus II 的一般使用方法外，增加了部分实例，并对宏功能模块（又称 IP 核）的使用进行了介绍，删除了 MAX + plus II 的相关内容。

全书共分为 4 章。

绪论简单介绍数字系统设计的基本概念、数字系统设计方法、可编程逻辑器件、EDA 软件的种类及各自的特点、硬件描述语言以及本课程的教学要求。

第 1 章为 Quartus II 使用练习，通过一些简单的例题，学生可以掌握 Quartus II 软件的使用方法。在本章的内容中，将出现一些简单的由 VHDL 或 Verilog HDL 进行的设计实例，学生完全可以通过这些简单例题掌握一般问题的硬件描述语言设计，若需要对硬件描述语言有进一步的了解，可以参考第 2 章和第 3 章的有关内容。对所有本章介绍的实例都给出了 VHDL 和 Verilog HDL 源程序，以供学生使用。

第 2 章介绍 VHDL，以尽量简练的方法介绍 VHDL 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 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 设计程序和仿真结果。本章的仿真基于 Quartus II。

第 3 章介绍 Verilog HDL，按照 IEEE 1364—1995 标准，介绍 Verilog HDL 的基础概念、语言要素、基本语句及仿真验证等，并给出设计实例。本章的仿真基于 ModelSim。

第 4 章介绍数字系统设计若干题目，分别给出设计要求、设计提示和设计框图，要求学生根据前 3 章的内容，使用 Quartus II 仿真软件，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 A 给出了第 4 章的部分数字系统设计题目的 VHDL 参考代码。

附录 B 给出了第 4 章的部分数字系统设计题目的 Verilog HDL 参考代码。

本书的绪论、第 1 章、第 4 章、附录 A、附录 B 由李国丽编写；第 2 章、第 3 章由朱维勇编写。王秀芹对书中例题做了大量验证工作。

解放军电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，参考了有关专家的著作，在此一并表示衷心的感谢！

由于编者水平有限，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	2.2.1 VHDL 的标记	97
0.1 数字系统设计的基本概念	1	2.2.2 VHDL 的数据类型	98
0.2 数字系统设计方法简介	2	2.2.3 VHDL 的运算符	103
0.3 可编程逻辑器件简介	3	2.2.4 VHDL 的数据对象	107
0.4 EDA 软件种类及各自特点	5	2.3 VHDL 设计的基本语句	111
0.4.1 电子电路设计与仿真工具	5	2.3.1 并行信号赋值语句	111
0.4.2 PCB 设计工具	6	2.3.2 条件赋值语句——WHEN - ELSE ...	112
0.4.3 集成电路设计工具	7	2.3.3 选择信号赋值语句——WITH -	
0.4.4 PLD 设计工具	8	SELECT	112
0.5 硬件描述语言简介	8	2.3.4 块语句	113
第1章 Quartus II 使用练习	10	2.3.5 IF - ELSE 语句	114
1.1 Quartus II 概述	10	2.3.6 CASE - WHEN 语句	116
1.2 基于 Quartus II 的电路设计过程	11	2.3.7 FOR - LOOP 语句	117
1.3 计数器设计	25	2.4 VHDL 高级语句	118
1.4 扫描显示电路	29	2.4.1 进程语句	118
1.5 数字系统设计实例	39	2.4.2 元件定义语句和元件例化语句 ...	120
1.5.1 具有 3 种信号灯的交通灯		2.4.3 生成语句	121
控制器	39	2.4.4 子程序	124
1.5.2 4 位数字频率计	46	2.4.5 程序包的设计	128
1.5.3 4×4 矩阵键盘扫描和按键		2.5 VHDL 设计实例	128
编码显示电路	52	2.5.1 常见的组合逻辑电路设计	128
1.5.4 基于 TLC549 的数字电压表	60	2.5.2 常见的时序逻辑电路设计	134
1.5.5 基于 TLC5620 的三角波		2.5.3 状态机设计	144
产生电路	68	第3章 Verilog HDL	148
1.6 宏功能模块及其应用	75	3.1 Verilog HDL 概述	148
1.6.1 宏功能模块的分类	75	3.1.1 Verilog HDL 的特点	148
1.6.2 宏功能模块的配置方法	77	3.1.2 Verilog HDL 的基本结构	148
1.6.3 宏功能模块使用实例	81	3.2 Verilog HDL 的语言要素	149
第2章 VHDL	91	3.2.1 基本语法定义	149
2.1 VHDL 概述	91	3.2.2 数据类型	150
2.1.1 VHDL 的特点	91	3.2.3 运算符	155
2.1.2 VHDL 的基本结构	91	3.2.4 Verilog HDL 编译向导	160
2.1.3 VHDL 的实体说明语句	93	3.2.5 系统任务与系统函数	162
2.1.4 VHDL 的结构体	94	3.3 Verilog HDL 基本语句	166
2.1.5 程序包、库和 USE 语句	95	3.3.1 赋值语句	166
2.2 VHDL 的数据类型和数据对象	97	3.3.2 条件语句	170

3.3.3 循环语句	175	4.9 乒乓球比赛游戏机	232
3.3.4 块语句	178	4.9.1 设计要求	232
3.3.5 结构化语句	179	4.9.2 设计提示	233
3.3.6 任务与函数	181	4.10 具有4种信号灯的 交通灯控制器	234
3.4 Verilog HDL 门元件和 结构描述	183	4.10.1 设计要求	234
3.4.1 门元件	184	4.10.2 设计提示	235
3.4.2 门级结构描述举例	187	4.11 出租车自动计费器	236
3.4.3 Verilog HDL 程序设计的 描述方式	188	4.11.1 设计要求	236
3.5 仿真验证	190	4.11.2 设计提示	237
3.6 可综合性描述	202	4.12 自动售邮票机	237
3.7 设计实例	206	4.12.1 设计要求	237
3.7.1 译码电路	206	4.12.2 设计提示	238
3.7.2 编码电路	207	4.13 电梯控制器	239
3.7.3 数据分配器	208	4.13.1 设计要求	239
3.7.4 同步计数器	210	4.13.2 设计提示	239
3.7.5 移位寄存器	212	附录	241
3.7.6 有限状态机的设计	213	附录 A 部分数字系统设计 VHDL	
3.7.7 复杂逻辑电路设计	217	参考代码	241
第4章 数字系统设计题目	219	A.1 多功能数字钟主控电路	241
4.1 多功能数字钟的设计	219	A.2 数字式竞赛抢答器主控电路	248
4.1.1 设计要求	219	A.3 洗衣机控制器主控电路	252
4.1.2 设计提示	219	A.4 电子密码锁主控电路	254
4.2 数字式竞赛抢答器	221	A.5 乘法器主控电路	257
4.2.1 设计要求	221	A.6 乒乓球比赛游戏机主控电路	258
4.2.2 设计提示	221	A.7 具有4种信号灯的 交通灯 控制器主控电路	261
4.3 数字频率计	222	A.8 出租车自动计费器主控电路	264
4.3.1 设计要求	222	A.9 自动售邮票机主控电路	266
4.3.2 设计提示	222	A.10 电梯控制器主控电路	269
4.4 拔河游戏机	224	附录 B 部分数字系统设计 Verilog HDL	
4.4.1 设计要求	224	参考代码	272
4.4.2 设计提示	225	B.1 多功能数字钟主控电路	272
4.5 洗衣机控制器	225	B.2 数字式竞赛抢答器主控电路	277
4.5.1 设计要求	225	B.3 洗衣机控制器主控电路	283
4.5.2 设计提示	226	B.4 电子密码锁主控电路	286
4.6 电子密码锁	228	B.5 乘法器主控电路	288
4.6.1 设计要求	228	B.6 乒乓球比赛游戏机主控电路	292
4.6.2 设计提示	228	B.7 具有4种信号灯的 交通灯 控制器 主控电路	294
4.7 脉冲按键电话按键显示器	229	B.8 出租车自动计费器主控电路	297
4.7.1 设计要求	229	B.9 自动售邮票机主控电路	300
4.7.2 设计提示	230	B.10 电梯控制器主控电路	303
4.8 乘法器	231	参考文献	307
4.8.1 设计要求	231		
4.8.2 设计提示	231		

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和生产的各个领域, 以及人们日常生活的方方面面。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 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 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

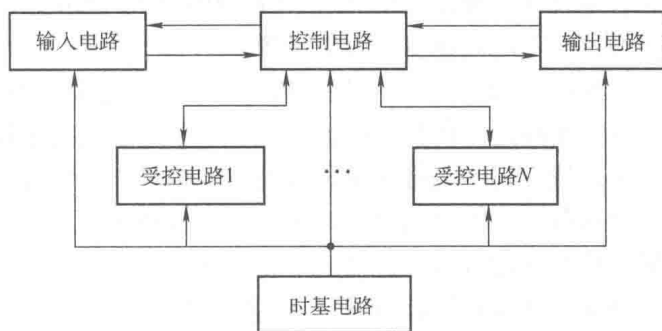


图 0-1 数字系统框图

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。

数字系统和功能部件之间的区别之一在于功能是否单一, 即一个存储器, 尽管规模很大, 可以达到数兆字节 (MB) 甚至吉字节 (GB), 但因其功能单一, 只能算是逻辑部件, 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, 即一个数字电路, 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如下。

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，应该对不同方案的性能、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；另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和复杂度的不断提高，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（Hard Description Language，硬件描述语言）的产生使自顶向下的设计方法得以实现。

自顶向下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子系统，再将每个子系统分成若干功能模块，再将每个模块分成若干小的模块……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，而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功能电路，设计起来并不困难。这样交通灯控制器设计的主要问题就是控制电路的设计，而这是一个规模不大的时序电路，这样就把一个复杂的数字系统的设计变成了一个较小规模的时序电路的设计，从而大大简化了设计的难度，缩短了设计周期。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如下。

- 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. 数字系统设计

数字系统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，因此最大限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（Complex Programmable Logic Device，复杂可编程逻辑器件）和 FPGA（Field Programmable Gate Array，现场可编程门阵列）两者的功能基本相同，只是实现原理略有不同，

所以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，或是 HDL 自由地设计一个数字系统，通过软件仿真，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系统，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 年代以后得到飞速发展，同时也大大推动了 EDA 软件和 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（E²PROM）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，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（Programmable array Logic, PAL）器件。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）器件，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 E²CMOS 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 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 CPLD。CPLD 是在 EPLD 的基础上发展起来的, 它采用 E²CMOS 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 E²PROM 的可编程逻辑器件叫 FPGA; 把基于乘积项技术、Flash 工艺 (类似 E²PROM 工艺) 的可编程逻辑器件叫 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, E²PROM 工艺)、FLEX 系列 (查找表技术, SRAM 工艺) 都叫作 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, 叫作 FPGA。

目前世界上有十几家生产 CPLD/FPGA 的公司, 最有代表性的是 Altera、Xilinx、Actel 和 Lattice。Altera 的主要产品有 Stratix 10、Stratix V、Arria 10、Arria V、MAX10、MAX V、Cyclone 等; Xilinx 的主要产品有 Spartan -6、Artix -7、Zynq、Kintex -7、Virtex -7、Kintex UltraScale、Kintex UltraScale +、Virtex UltraScale、Virtex UltraScale + 等; Actel 的主要产品有 ProASIC3、IGLOO、IGLOO2、SmartFusion、Fusion 等; Lattice 的主要产品有 ICE40 Ultra/UltraLite、ICE40 LP/HX/LM、Machx03、LatticeXP2、ispMach4000ZE 等。

0.4 EDA 软件种类及各自特点

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, 种类繁多, 按照主要功能和应用场合, EDA 工具可以大致分为以下 4 类。

0.4.1 电子电路设计与仿真工具

目前, 在我国各高校教学中具有广泛影响的电子电路设计与仿真工具软件有 SPICE、PSPICE、EWB、Multisim、MATLAB 等。

1. SPICE 与 PSPICE

SPICE (Simulation Program with Integrated Circuit Emphasis) 软件于 1972 年由美国加州大学伯克利分校的计算机辅助设计小组利用 FORTRAN 语言开发而成, 主要用于大规模集成电路的计算机辅助设计。SPICE 的正式实用版 SPICE 2G 在 1975 年正式推出, 但是该程序的运行环境至少为小型机。1985 年, 加州大学伯克利分校用 C 语言对 SPICE 软件进行了改写, 1988 年 SPICE 被定为美国国家工业标准。与此同时, 各种以 SPICE 为核心的商用模拟电路仿真软件, 在 SPICE 的基础上做了大量实用化工作, 从而使 SPICE 成为最为流行的电子电路仿真软件。

PSPICE 则是由美国 Microsim 公司在 SPICE 2G 版本的基础上升级并用于个人计算机上的 SPICE 版本, 其中采用自由格式语言的 5.0 版本, 自 20 世纪 80 年代以来在我国得到广泛应用, 并且从 6.0 版本开始引入图形界面。1998 年著名的 EDA 商业软件开发商 ORCAD 公司与 Microsim 公司正式合并, 自此 Microsim 公司的 PSPICE 产品正式并入 ORCAD 公司的商业 EDA 系统, 由 ORCAD 公司正式推出了 ORCAD PSPICE Release 9.0 等。

ORCAD PSPICE 具有如下特征: 在对模拟电路进行直流、交流和瞬态等基本电路特性分析的基础上, 实现了蒙特卡罗分析、最坏情况分析、温度和应力分析以及性能优化分析等较

为复杂的电路特性分析；不但能够对模拟电路进行，而且能够对数字电路、数/模混合电路进行仿真；具有强大的波形观察、分析及后处理表达功能；具有超越常规电路仿真的卓越能力，如事件驱动仿真、检查点重新启动、高级收敛性和曲线拟合等；拥有超过 33 000 个模拟/混合信号装置、数学函数及行为建模的仿真方案，使软件的仿真速度更快。

2. EWB

EWB (Electronic 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。

在 EWB 的桌面上提供了万用表、示波器、信号发生器、逻辑分析仪、数字信号发生器、逻辑转换器、波特图仪、电压表、电流表等仪器仪表。在它的器件库中提供了各种建模精确的元器件，如电阻、电容、电感、晶体管、二极管、继电器、可控硅、数码管等；各种运算放大器；74 系列 TTL 集成电路、4000 系列 CMOS 集成电路等。器件库中没有的元器件，可以由外部模块导入。

EWB 的兼容性很好，其文件格式可以导出为能被 OrCAD 或 Protel 读取的格式。

EWB 的升级版本 Multisim 2001 除具备上述功能外，还支持 VHDL 和 Verilog HDL 文本的输入。在此基础上，Multisim 经历了 Multisim 7、Multisim 8，后来 Multisim 被美国 NI (National Instrument) 公司收购，以后推出了 NI Multisim 9，其性能得到了极大的提升。最新推出的 NI Multisim 14.0 采用全新的主动分析模式，具备可视化效果极强的电压、电流和功率探针，可搭建先进的电源电路，可以与 MPLAB 进行协同仿真，Ultiboard 新增了 Gerber 和 PCB 制造文件导出函数。Multisim 还推出了 iPad 版本，使用者可以随时随地进行各种电路的仿真。

3. MATLAB

MATLAB (Matrix Laboratory) 是由美国 Mathworks 公司出品的数学计算、系统仿真和设计工具。Mathworks 公司拥有包括 MATLAB、Simulink、Polyspace 三个产品家族。MATLAB 是一种用于算法开发、数据可视化、数据分析以及数值计算的科学计算语言和编程环境。它提供了基本的数学算法，如矩阵运算、数值分析算法等，并集成了 2D 和 3D 图形功能，以完成相应数值可视化工作。Simulink 是一种用于对多领域动态和嵌入式系统进行仿真和模型设计的图形化环境，可以针对任何能够用数学来描述的系统进行建模，包括连续、离散、条件执行、事件驱动、单速率、多速率和混杂系统等。

0.4.2 PCB 设计工具

PCB (印制电路板) 设计的工具软件很多，主要有 Protel、OrCAD、PowerPCB、Candence PSD、Mentor Graphics 的 Expedition 系列、Zuken 的 CadStart、PCB Studio、TANGO 等。

1. Protel

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。早期的 Protel 主要作为 PCB 自动布线工具使用，经过多年的发展，功能越来越完善，现在已经是完整的全方位电路设计系统，它可以完成电路原理图的设计和绘制、模拟电路与数字电路混合信号仿真、多层 PCB 设计、

自动布局布线、可编程逻辑器件设计、图表生成、电路表格生成以及支持宏操作等。Protel 还兼容一些其他软件的文件格式，如 OrCAD、PSPICE、Excel 等。使用多层 PCB 的自动布线，可以实现高密度 PCB 的 100% 布通率。

Protel 在 2002 年推出 Protel 99se，随后是 DXP2002、DXP2004，2005 年底，Protel 系列推出了的新版本 Altium Designer 6，它除了全面继承包括 Protel 99se、Protel DXP 在内的先前一系列版本的功能和优点以外，还增加了许多改进和很多高端功能，拓宽了板级设计的传统界限，全面集成了 FPGA 设计功能和 SOPC 设计实现功能，从而允许工程师将系统设计中的 FPGA 与 PCB 设计及嵌入式设计集成在一起。

2. OrCAD

OrCAD 是由 OrCAD 公司于 20 世纪 80 年代末推出的 EDA 工具，它集成了原理图绘制、PCB 设计、数字电路仿真、可编程逻辑器件设计等功能，界面友好直观，元器件库丰富。2003 年，OrCAD 与 Cadence 公司合并后，Cadence OrCAD 具有了模拟与数字电路混合仿真功能，元器件库达到 8 500 个，收录了几乎所有的通用型电子元器件模块。最新版本的 OrCAD 16.6 让 PCB 的设计进入更细节阶段，它包括设计输入工具 OrCAD Capture，模拟与混合信号仿真工具 OrCAD PSpice Designer，电路板设计工具 OrCAD PCB Designer、OrCAD PCB SI 等，是世界上使用最广泛的 EDA 工具之一。

0.4.3 集成电路设计工具

集成电路 (Integrated Circuit, IC) 设计工具主要被 Cadence、Synopsys 和 Mentor 等软件供应商所垄断，大致可以分为以下几种。

1. 设计输入工具

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 器件厂商一般都有自己的设计输入编辑器，如 Cadence 的 Compoaer、Viewlogic 的 Viewdraw，但其设计 FPGA/CPLD 的工具大都可以作为 IC 的设计输入工具，如 Xilinx 的 Foundation、ISE，Altera 的 Quartus II 和 Modelsim FPGA 等。

2. 仿真工具

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。

几乎每个公司的 EDA 产品都有仿真工具。Verilog - XL、NC - verilog 用于 Verilog 仿真，Leapfrog 用于 VHDL 仿真，Analog Artist 用于模拟电路仿真。Viewlogic 的仿真器有 Viewsim 门级电路仿真器、Speedwave VHDL 仿真器、VCS - verilog 仿真器。Mentor Graphics 有其子公司 Model Tech 出品的 VHDL/Verilog 双仿真器 ModelSim。Cadence、Synopsys 用的是 VSS (VHDL 仿真器)。现在的趋势是各大 EDA 公司都逐渐用 HDL 仿真器作为电路验证的工具。

3. 综合工具

综合工具可以把 HDL 变成门级网表。这方面 Synopsys 工具占有较大的优势，它的 Design Compile 是作为一个综合的工业标准，它还有另外一个产品叫 Behavior Compiler，可以提供更高级的综合。

随着 FPGA 设计的规模越来越大，各 EDA 公司相继开发了用于 FPGA 设计的综合工具，

比较有名的有：Synopsys 的 FPGA Express、Cadence 的 Synplity、Mentor 的 Leonardo。这三家的 FPGA 综合软件占了市场的绝大部分份额。

4. 布局布线工具

布局布线工具又称适配器，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栽文件，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。

在 IC 设计的布局布线工具中，Cadence 的软件比较强，有很多产品，最有名的是 Cadence Spectra。它原来是用于 PCB 布线的，后来 Cadence 把它用来作 IC 的布线。其主要工具有：Cell3、Silicon Ensemble - 标准单元布线器、Gate Ensemble - 门阵列布线器、Design Planner - 布局工具。其他各 EDA 软件开发公司也提供各自的布局布线工具。

5. 物理验证工具

物理验证工具包括版图设计工具、版图验证工具、版图提取工具等。这方面 Cadence 很强，其 Dracula、Virtuoso、Vampire 等物理验证工具有很多人使用。

在 ASIC 设计中，验证工作占了相当大的比例，尤其是进入深亚微米领域后，验证工作花去几乎 80% 的时间，平均验证成本已达总设计费用的 70%，所以有实力的 EDA 厂商都非常重视验证工具的开发，尤其是开放式统一验证平台的开发。

6. 模拟电路仿真工具

对于模拟电路的仿真，普遍使用 SPICE，不同的公司有自己的 SPICE，如 MiceoSim 的 PSPICE、Meta soft 的 HSPICE 等（已被 Avanti 公司收购）。

0.4.4 PLD 设计工具

可编程逻辑器件的基本设计方法借助 EDA 工具软件，用原理图、状态机、布尔表达式、硬件描述语言等方式进行设计输入，仿真后形成目标文件，经过编程器或下载电缆，由目标器件实现。只要有数字电路基础，会使用计算机，借助 EDA 工具就可以进行 PLD 的开发。工程师可以在几分钟内完成以往需要几周才能完成的工作，并可数将数百万门电路的复杂设计集成在一块芯片中，并且 PLD 器件的在线可编程技术使得硬件的修改非常方便。目前，PLD 已经成为现代数字系统的主要设计手段，是电子工程师必备的设计技术。

随着可编程逻辑器件应用的日益广泛，制造工艺的不断提高，PLD 芯片不仅提供 100 万以上的逻辑门，还提供丰富的 IP 芯核库，可以缩短开发时间，降低成本。

世界上各大可编程逻辑器件生产厂商都有自己的 EDA 开发平台。例如，Lattice 的 Lattice Diamond 和 Synario、Xilinx 的 Vivado Design Suite 和 ISE、Altera 的 Quartus Prime 和 Quartus II 等，都是得到广泛使用的开发平台。

随着器件规模的不断增加，软件的复杂性也随之提高。所以，由专门的软件公司与器件供应商合作，各种功能强大的设计软件将不断被开发出来。

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已有 20 多年的历史,并成功地应用于数字系统开发的设计、综合、仿真和验证等各个阶段,使设计过程达到高度自动化。硬件描述语言有多种类型,最具代表性和使用最广泛的是 VHDL (Very High Speed Integrated Circuit Hardware Description Language, 超高速集成电路硬件描述语言)和 Verilog HDL。

VHDL 于 20 世纪 80 年代初由美国国防部(The United States Department of Defense)发起创建,当时制订了一个名为 VHSIC (Very High Speed Integrated Circuit, 超高速集成电路)的计划,其目的是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 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协会(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 最初于 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。由于其模拟、仿真器产品的广泛使用,Verilog HDL 作为一种便于使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 Verilog 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 Quartus II 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题目,期望通过实例和练习,把数字系统设计的基本理论、基本方法和设计题目紧密结合,使读者在 Quartus II 的设计平台下,学会用原理电路图输入或硬件描述语言(VHDL 或 Verilog HDL) 输入进行电路设计、编译(Compiler)、仿真(Simulator)、底层编辑(Floorplan Editor)及 PLD 器件编程校验(Programmer 或 Configure),但对功能测试向量(Waveform Editor)、逻辑综合与试配(Logic Synthesize)等涉及不多,以提高读者利用 Quartus II 进行数字系统设计的能力。

第 1 章 Quartus II 使用练习

1.1 Quartus II 概述

Quartus II 是 Altera 公司自行设计的 CAE 软件平台，提供了完整的多平台设计环境，能满足各种特定设计的要求，是单片可编程系统（SOPC）设计的综合性环境和 SOPC 开发的基本设计工具，并为 Altera DSP 开发包进行系统模型设计提供了集成综合环境。Quartus II 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。

Quartus II 开发系统具有下列诸多特点。

1. 界面开放

Quartus II 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 工具设计输入工具来建立逻辑设计，用 Quartus II 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，Quartus II 支持与 Cadence、Exemplarlogic、Mentor Graphics、Synopsys、Synplcity、Viewlogic 等公司所提供的 EDA 工具接口。

2. 与结构无关

Quartus II 系统的核心编译器支持 Altera 公司的 FLEX10K、FLEX8000、FLEX6000、MAX9000、MAX7000、MAX5000、Classic、Stratix、Stratix II、Cyclone、Cyclone II 等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。Quartus II 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

Quartus II 提供丰富的库单元供设计者调用，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。此外，Quartus II 含有许多用来构建复杂系统的参数化宏功能模块和 LPM（Library of Parameterized Modules）模块，它们可与 Quartus II 普通设计文件一起使用，使非专业设计人员完成 SOPC 设计成为可能。

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 Quartus II 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，本书侧重点在于用 Quartus II 进行各种设计输入（图形或 HDL 输入）、编译、仿真、底层编辑及 PLD 器件编程校验，并不过多涉及功能测试向量、逻辑综合与试配等。

5. 硬件描述语言

Quartus II 软件支持各种 HDL 设计输入选项，包括 VHDL、Verilog HDL 和 AHDL。Quartus II 内部嵌有 VHDL、Verilog 及 AHDL 的逻辑综合器，也可调用第三方的综合工具，如 Le-

onardo Spectrum、Synplify Pro、FPGA Compiler II 等进行逻辑综合。

本章将通过一些例子来说明利用 Quartus II 进行数字功能模块或数字系统设计的过程。

1.2 基于 Quartus II 的电路设计过程

Quartus II 中每一项设计都对应一个工程 (Project)。Quartus II 软件中的工程由所有设计文件和与设计文件有关的设置组成。为便于设计项目的存储, 必须首先建立一个存放与此工程相关的所有文件的文件夹, 如 E:\quartus2\test。此文件夹被默认为工作库 (Work Library)。一般, 不同设计项目应该放在不同文件夹中。

【例 1-2-1】用原理图输入法设计一个 3 线 -8 线译码器。

步骤 1: 进入 Windows 操作系统, 打开 Quartus II 软件。

步骤 2: 创建工程。

(1) 工程设置。选择 File → New Project Wizard 菜单命令, 在弹出的图 1-2-1 所示的对话框中, 单击 Next 按钮, 进入图 1-2-2 所示的工程设置界面, 设置工程文件夹、工程名和顶层实体名。工程文件夹、工程名和顶层实体名以字母和数字的组合命名, 不可以是中文, 也不可以是元件库中已有的元件名, 如 and2、input 等。在图 1-2-2 中设置工程名和顶层实体名均为 L38。设置完成后单击 Next 按钮, 进入目标器件选择界面。

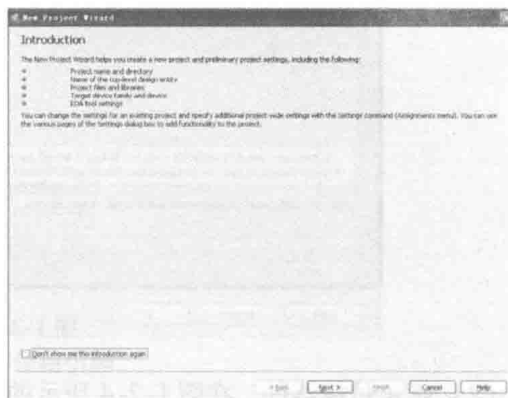


图 1-2-1 新建工程

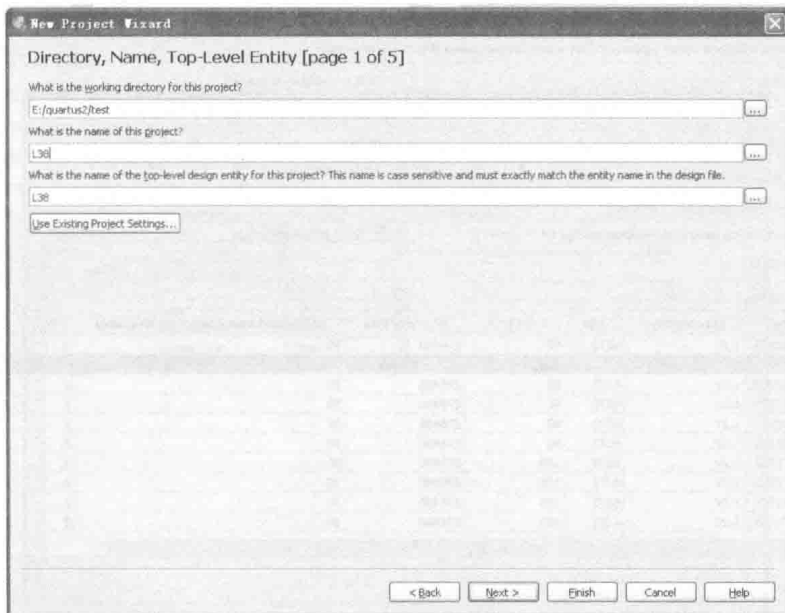


图 1-2-2 工程设置