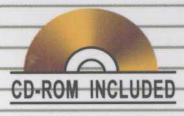
“十二五”高等院校规划教材



内含源代码和课件

CPLD/FPGA 设计与应用高级教程

郭利文　邓月明　编著
黄智伟　主审

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

“十二五”高等院校规划教材

CPLD/FPGA 设计与应用高级教程

郭利文　邓月明　编著
黄智伟　主审

北京航空航天大学出版社

内 容 简 介

本书结合目前主流的CPLD/FPGA产品以及最流行的设计理念，系统、详细地介绍CPLD/FPGA的硬件结构、硬件描述语言与验证语言的基础应用以及高级应用；详细介绍如何使用Verilog HDL语言进行有限状态机设计和testbench设计，以及如何使用Modelsim进行功能仿真和时序仿真；简要介绍验证方法学的基本概念以及验证语言的比较，并就CPLD/FPGA的系统应用进行了详细探讨，包括DSP设计、嵌入式处理器设计、HardCopy设计、嵌入式逻辑分析仪的使用以及CPLD/FPGA的板级设计。

本书既可作为电子信息、通信工程以及相关工科专业的本科高年级学生和研究生教材，也可作为全国大学生电子设计竞赛的培训教材，以及从事电子电路系统设计与CPLD/FPGA/ASIC设计的工程技术人员的参考用书。

图书在版编目(CIP)数据

CPLD/FPGA设计与应用高级教程 / 郭利文，邓月明编著. --北京 ：北京航空航天大学出版社，2011.1

ISBN 978-7-5124-0246-1

Ⅰ. ①C… Ⅱ. ①郭… ②邓… Ⅲ. ①可编程序逻辑器件—教材 Ⅳ. ①TP332.1

中国版本图书馆CIP数据核字(2010)第209231号

CPLD/FPGA设计与应用高级教程

郭利文　邓月明　编著

黄智伟　主审

责任编辑　冯　颖

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(邮编100191)　http://www.buaapress.com.cn

发行部电话：(010)82317024　传真：(010)82328026

读者信箱：bhpress@263.net　邮购电话：(010)82316936

北京市松源印刷有限公司印装　各地书店经销

*

开本：787×1092　1/16　印张：20　字数：512千字

2011年1月第1版　2011年1月第1次印刷　印数：4 000册

ISBN 978-7-5124-0246-1　定价：39.00元(含光盘1张)

前　言

本书基于当前主流的CPLD/FPGA器件及最流行的设计理念，根据作者多年来的实际设计经验，系统介绍了CPLD/FPGA的硬件结构、硬件描述语言与硬件验证语言的基础应用以及高级应用，详细介绍了如何在工程中利用Verilog HDL语言进行有限状态机设计和Testbench设计，以及如何使用Modelsim进行功能仿真和时序仿真；通过相关实例具体阐述了怎样实现SOPC的设计以及Nios II的应用开发，并就CPLD/FPGA的系统应用进行了详细探讨，包括DSP设计、嵌入式处理器设计、HardCopy设计、嵌入式逻辑分析仪的使用以及CPLD/FPGA的板级设计，从而很好地满足了可编程逻辑器件工程应用整个流程的知识需要。全书实例丰富、图文并茂，由浅入深、由易到难详细介绍了CPLD/FPGA的设计与应用。

全书共分11章：第1章简要介绍了CPLD/FPGA的发展历程、设计语言与设计方法、主要产品以及前景展望，介绍了硬件语言和软件语言的区别与联系，重点阐述了CPLD/FPGA的设计、验证流程及其注意事项；第2章分别从传统和最新CPLD/FPGA的硬件结构阐述了乘积项和查找表的基本原理以及CPLD/FPGA的选择指导，重点讲述了最新CPLD/FPGA的硬件结构(这是本书紧扣最新技术发展的体现之一)；第3、4章主要讲述了目前工程界中使用最为广泛的Verilog HDL硬件描述语言，从工程实践的角度具体介绍了它的基本语法及其主要特点，重点讲述了Verilog HDL语言的高级编程应用以及参数化设计；第5章通过实例说明着重阐述了有限状态机的特点、设计和使用——包括目前最流行的设计方法；第6章简单地介绍了约束的基本概念以及约束的方式应用，重点阐述了在CPLD/FPGA设计中的时延设计与分析；第7章是对前面几章的概括，总结了采用硬件描述语言进行CPLD/FPGA设计时的一些基本设计原则和技巧，特别提出了组合逻辑和时序逻辑设计设计时的注意事项，以及代码风格的重要性；第8章主要讲述了仿真原理、Testbench设计以及如何通过采用Modelsim软件来实现仿真；第9章主要就验证与测试进行了探讨，简要介绍了HVL(硬件验证语言)和断言的重要性；第10章就CPLD/FPGA的高级应用(包括DSP、嵌入式CPU、嵌入式逻辑分析仪等方面的知识)进行分析阐述，通过相关的实例详细说明了DSP和SoPC的相关应用；第11章是对整个CPLD/FPGA设计的总结，重点阐述了CPLD/FPGA的系统应用，包括信号完整性、电源完整性、功耗与热设计、PCB设计等方面的内容。

与其他教材相比，本书的主要特点体现在以下5个方面：

① 内容新　知识体系以及技术设计理论都是近年来最新、最常用的，实例所使用的实验软硬件平台也都是最近两年三大公司的主流平台；

② 技术实用　全书侧重实例讲解以及应用设计，其中的实例都是从工程实践中提取的，读者可直接使用；

③ 适用面广　一方面指读者面广，适用于高校学生和工程技术人员，另一方面是指教材中的程序实例大部分适用于3个主流可编程逻辑器件公司的软硬件平台（个别实例是平台相关的，已在书中特别指出）；

④ 适合教学　通过加入大量的短小实例使读者及时理解消化所讲的理论知识点，保证知识点教与学的完整性，同时在每章最后一节加入一个较完整的实例，使得读者能对本章内容进行系统性的掌握和应用；

⑤ 知识点全面　将产品开发所需的基础知识、编程技巧、开发调试、验证测试及最终系统设计的整套技术流程都进行了介绍，特别是CPLD/FPGA的验证和系统开发方面的知识目前大多数CPLD/FPGA书籍中很少涉及，但对于系统设计来说又是必不可少的部分，这也是本书的重点和主要特色之一。

本书另附配套光盘1张，提供了本书的多媒体课件以及书中所有示例的设计源代码，供读者参考。

全书由郭利文、邓月明编写，由黄智伟教授主审。林韦成（中国台湾）、王康斌、高芳莉和邱树林等工程师为本书的编写付出了诸多努力，提供了许多详细的建议和第一手工程实践资料；Tomonori Hirai博士（美国）、Jyhming Zhong博士（美国）和Mario Lee博士（美国）等在信号完整性、高速信号与PCB设计、功耗与热设计等方面给予了大量的指导；成都电子科技大学楚佩斯、湖南师范大学胡强、贺洪平、王赟、李青、刘斌、欧阳亚、李慧琳、张谦、袁纯、潘律、唐波以及中南大学肖成等同学也为本书的编写做了大量的工作，从而促成了本书的迅速问世。在此一并表示衷心的感谢。同时还要感谢湖南省师范大学青年基金项目课题组（编号：71003）和教学改革研究项目课题组对本书出版的支持。

在本书的编写过程中，作者参考了大量的国内外著作和资料，听取了多方面的宝贵意见和建议，在此对他们致以衷心的感谢。

由于作者水平有限，书中错误和不足之处在所难免，敬请各位读者批评斧正。

作　者

2011年1月

目 录

第 1 章

概　述

本章重点介绍数字电路设计的建模以及可编程逻辑器件的一些基本概念，简单介绍CPLD/FPGA的设计理念及其验证流程。

本章主要内容如下：

- 数字电路的发展演变；
- CPLD/FPGA的介绍；
- 设计语言及其方法的介绍；
- 硬件语言与软件语言的区别；
- 设计与验证流程；
- CPLD/FPGA的前景与展望。

1.1　数字电路基础及发展演变

1904年英国科学家弗莱明为自己发明的电子管弗莱明“阀”申请了专利，人类历史上第一只电子管诞生了，世界从此迈入了电子时代。图1-1就是当时较为流行的一款电子管。1906年德福雷斯特在此基础上发明了三极管，从此三极管成为了被广泛应用的电子器件。而今随着电子技术的不断发展，数字电路相继从晶体管、中小规模集成电路、超大规模集成电路(VLSIC)发展到今天的专用集成电路(ASIC)，数字逻辑器件也由从前简单的逻辑门发展到了复杂的SOC(System On Chip)。

随着科学技术及其工艺的发展，体积小、功耗低、集成度高成为了IC设计所追求的目标。系统的逻辑复杂度、规模与日俱增，日新月异。20世纪60年代，Gordon Moore预测一块芯片上集成的晶体管的数量将呈指数规律增长，即所谓的“摩尔定律”。他认为芯片上的晶体管的集成度每18个月将会翻一番，换一句话来说，就是在单位面积上逻辑门的数量每18个月翻一番，即呈指数级数增长，累计到现在已经超过2^{30}。到目前为止，这个定律依然正确。目前一个Levin超大规模的集成电路芯片，集成度一般达到了十几亿门。而这样高的集成度也要求人

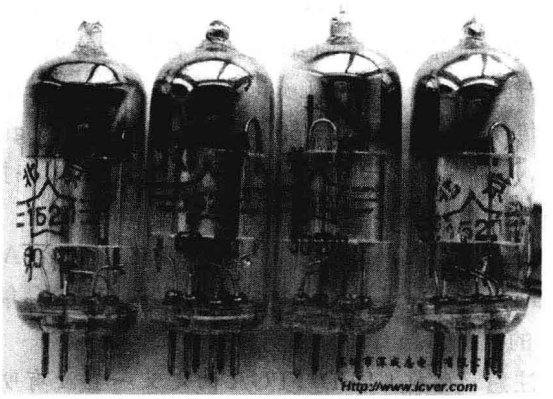


图1-1　电子管

们思考：该把芯片设计成固定功能，还是设计成可以灵活地选择功能呢？于是集成电路芯片便朝着两个方向并行发展，一个是专用集成芯片（Application Specific Integrated Circuit，ASIC），另一个是可编程逻辑器件。ASIC一般应用于固定场合。人们设计线路时，需要按照它的要求进行设计，这就要求电子工程师具有一定的技术和经验。另外，有些ASIC功能复杂且繁多，而实际上具体应用只需要其中的一部分，这样就造成了逻辑资源的浪费。而可编程逻辑器件功能强大，人们可以通过编程来实现自己想要的功能，去除不需要的冗余功能。只要了解编程软件和硬件的编程语言就可以实现对芯片的设计，这样可以加速芯片设计并缩短其上市时间（Time To Market，TTM）。

另一方面，这样复杂的集成度也就意味着传统的设计方法和设计理念已经不能适应芯片的产业发展，这就要求人们去变革和更新现有的设计理念以及设计方法。在早期简单的逻辑门设计阶段，电子辅助设计（EDA）的作用并不明显，人们往往习惯于使用卡诺图来简化设计，然后通过单板等试验进行系统设计验证。计算机的出现及发展带动和加速了软件产业的发展，从而使利用软件编程方法完成硬件芯片电路设计成为现实。这时，工程师开始使用EDA工具来对原理图进行仿真分析化简，分析其性能特征，可是因为原理图的可移植性差、维护起来费时费力的缺点，它只适用于一些比较简单的集成电路芯片，而当集成度复杂到ASIC和可编程逻辑器件时，一种抽象性更高、应用起来更简单的设计语言——硬件描述语言（Hardware Design Language，HDL）便应运而生。硬件描述语言的出现加速了芯片设计的速度，缩短了上市时间，而硬件描述语言本身也在日益丰富与壮大。

1.2 CPLD/FPGA的介绍

CPLD/FPGA是用户根据需要自行构造逻辑功能的数字集成电路。CPLD/FPGA基本设计方法就是借助集成开发软件平台，用画原理图或者硬件描述语言编程的方法生成相应的网表文件，然后通过布局布线以及时序约束下载到目标芯片中，实现设计的数字系统。

CPLD/FPGA的产生是集成电路的不断发展带动了计算机的高速发展，从而带动了软件技术更新、EDA技术飞速发展的结果。历史上，第一个可编程逻辑阵列器件PLA（Programmable Logic Array）出现在20世纪70年代中期，它的主要结构是由“与或”阵列组成的，与阵列和或阵列都可以同时编程。随着技术的发展，可编程逻辑器件已经从PLA发展到PAL（Programmable Array Logic）再到GAL（Generic Array Logic），目前已发展到CPLD/FPGA。

传统的CPLD（Complex Programmable Logic Device）发展来自于典型的PAL、GAL器件结构，任何一个组合逻辑可以用“与或”表达式来描述，能够实现大量的组合逻辑功能，适合于中小规模通用数字集成电路。

传统的FPGA（Field Programmable Gate Array）基于查找表结构，绝大多数都是基于SRAM的结构，相对而言集成度比CPLD要高，能够快速实现许多复杂的逻辑功能，甚至用于SOC的设计。

注意：目前CPLD/FPGA的定义一直没有明确，不同厂家的叫法也不尽相同。Xilinx公司把基于查找表技术、SRAM技术和要外挂配置用的E^2PROM的PLD叫作FPGA；把基于乘积项技术、Flash工艺的PLD叫作CPLD。Altera公司把自己的PLD产品——MAX系列（乘积项技术、Flash工艺）、FLEX系列（查找表技术，SRAM工艺）都叫作CPLD。事实上，随着技

术的发展，最新的CPLD和FPGA结构已经越来越相似，特别是Lattice公司的XO系列和Altera公司的MAXII系列问世后，人们很难用一个明确的定义去区分CPLD和FPGA。

经过几十年的发展，许多公司都开发出了CPLD/FPGA可编程逻辑器件，而经过行业的调整和整合，目前Altera、Lattice和Xilinx公司的CPLD/FPGA产品已经占到了市场份额的80%以上。

Altera公司在20世纪90年代以后发展迅猛，是最大的可编程器件供应商之一。在CPLD产品方面连续推出了MAX3000/7000、FLEX10、MAXII、MAXIIZ系列，而在FPGA产品方面也不断推陈出新，推出了Cyclone、Stratix、Arria等系列。

Xilinx公司是FPGA的发明者，是老牌的FPGA公司，也是最大的可编程逻辑器件供应商之一，其产品丰富，种类齐全，主要有XC9500、Coolrunner、CoolrunnerII等系列的CPLD产品以及Spartan、Virtex系列的FPGA产品，目前最新款的FPGA是Artix-7、Kinex-7，Virtex-7系列的产品。

Lattice公司是ISP(In System Program，在系统编程)技术的发明者，而这一项技术极大地促进了PLD产品的发展。在20世纪90年代末、21世纪初，相继收购了Vantis(原AMD子公司)和Agere公司(原Lucent微电子部)，从而成为了第三大可编程逻辑器件供应商。它可以提供从PAL到FPGA的一系列产品，甚至包括模拟器件，其中CPLD比较有特色，特别是XO系列。目前，该公司的主要的产品有ispMACH4000系列、XO系列CPLD、EC系列、XP系列FPGA，以及可编程模拟器件等。

通常而言，在欧美市场使用Xilinx的人较多，而在亚太地区使用Altera的人相对较多。全球60%的CPLD/FPGA产品都是由Altera/Xilinx公司提供的。

除此之外，目前世界上还有几家比较有影响的CPLD/FPGA公司，如Actel、Qlogic等，另外2008年新成立了一家CPLD/FPGA公司——SiliconBlue公司。不过短时间内，它们还无法与上述三家公司匹敌。

1.3 设计语言及其方法的介绍

随着计算机技术和EDA技术的不断发展，各家可编程逻辑器件公司开发了各种硬件描述语言，如AHDL(Analog Hardware Description Language，模拟硬件描述语言，Altera公司为自家产品而开发的)、ABEL(Advanced Boolean Equation Language即高级布尔方程语言，Lattice公司专为其产品而开发的)。

1981年美国国防部提出了一种新的标准化语言——VHDL(Very high speed integrated circuit Hardware Description Language，超高速集成电路硬件描述语言)。它的语法丰富，语句类型繁多，功能强大，结构严谨，并且能够在高层次上进行仿真描述，成为了最受欢迎的硬件描述语言之一，于1987年成为了IEEE标准。

1983年，美国GDA(Gateway Design Automatic)公司的Phi Moordy创立了Verilog HDL语言，并在1984～1985年期间设计出了仿真器VerilogXL，之后又于1986年提出了用于快速门级仿真的XL算法。1989年GDA公司被Cadence公司收购，1990年Cadence公司决定开发Verilog HDL语言，并成立了OVI(Open Verilog International)组织来促进Verilog HDL语言的发展。它吸收了VHDL和C语言的长处，有着类似于C语言的语法体系，设计

人员只要有C语言基础就可以入门。1995年,IEEE制定了Verilog HDL的IEEE标准即Verilog HDL1364-1995,之后又于2001年发布了Verilog1364-2001标准。

Verilog HDL和VHDL作为最流行的HDL,两者之间孰优孰劣一直是设计工程师不断争论的话题。二者在设计上都能胜任数字电路设计的任务,目前几乎所有的CPLD/FPGA设计综合和仿真软件平台都支持这两种语言。

硬件描述语言一直在这两种语言的基础上发展和演化着,之后又出现了许多更为抽象的硬件描述语言,如SystemVerilog、SystemC、Superlog和CoWare C等语言。这些高级HDL的语法结构更加丰富,更适合系统级、功能级等高层级的设计描述与仿真。图1-2所示为HDL适用层次示意图。

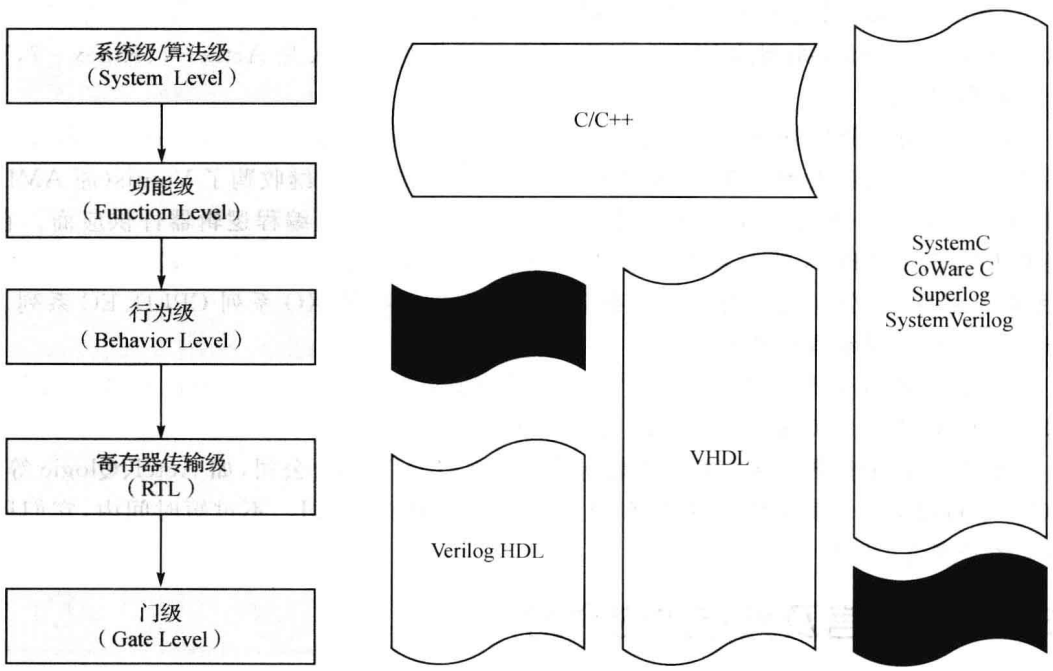


图1-2 设计语言与设计层次的关系

1.4 硬件语言与软件语言的区别

硬件语言与软件语言有着本质的区别。Verilog和VHDL语言作为硬件语言,其本质是在描述硬件。虽然从某种意义上来说,Verilog HDL借鉴了许多C语言的要素,可是它描述出来的结果就是芯片内部实际的电路,所以一段HDL代码的优劣就在于它转化成所描述电路性能是否合理和流畅,而非代码是否简洁。

硬件语言与软件语言相比较,最大的区别在于软件语言缺乏了以下3个基本概念。

① 互连(Connectivity):互连是硬件系统中的一个基本概念,而软件语言并没有这样的描述。Verilog HDL中的wire变量可以很好地表达这样的功能。

② 并发(Concurrency):软件语言天生就是串行的,只有执行了上一行程序才能进入下一行程序,这样与硬件系统的设计理念相违背的,硬件语言基于并行,能够有效地描述硬件系统。

③ 时序(Timing)：软件语言的运行没有一个严格的时间时序概念，程序运行的快慢取决于处理器本身的性能，而硬件语言可以通过时间度量与周期的关系来描述信号之间的关系。

但是硬件描述语言在抽象程度上比起软件语言相对差一些，语法也不如软件语言灵活，特别是文件的输入/输出。为了克服这些缺陷，PLI(Programmable Language Interface，编程语言接口)也就应运而生，这样就可以在仿真器里面实现C语言程序和Verilog HDL程序的相互通信，或者是Verilog HDL调用C语言的函数库，提高了Verilog HDL的灵活性和抽象能力。

1.5 设计与验证流程

当硬件系统进入软件编程设计阶段时，基本上之前所有传统的设计理念和手段都需要更新和变革。传统上采用的“自下向上(Bottom Up)”的设计方法已经不再适应CPLD/FPGA的设计需求。CPLD/FPGA的设计需要从系统整体要求出发，自上向下逐步将设计内容细化，最后完成系统硬件的整体设计，这就是所谓的“自顶向下(Top Down)”的设计方法，也就是从抽象到具体的一种设计方法。与传统的设计方法相比较，自顶向下的方法不仅节约了大量的设计时间，也大大减少了调试时间，整体上缩短了系统的设计时间，节省了大量的人力与物力，加速了产品上市。图1-3所示为HDL硬件设计与验证的流程，有些步骤可以根据系统的复杂程序进行适当的修剪、省略。

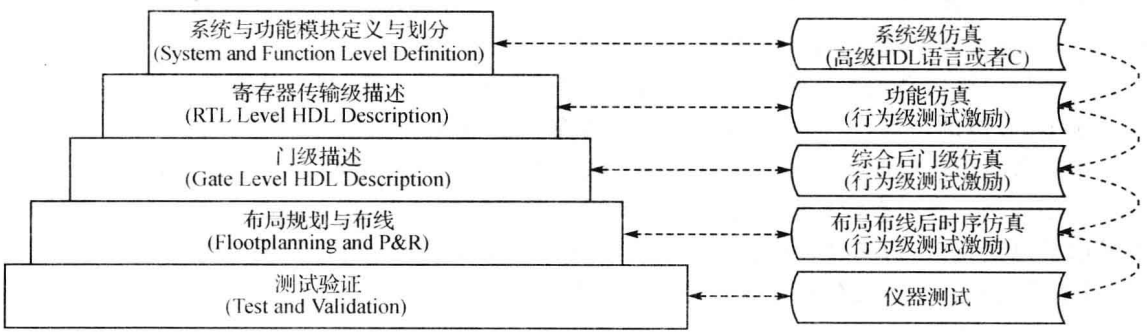


图1-3 设计层次及任务要求

1. 系统与功能模块定义

在大型系统的设计与视线中，首先需要对整个系统进行详细的系统规划与描述，并且通过系统级的仿真来整体权衡和考量系统的性能和功能实现。这时侧重于系统的规划与实现，而此时使用的仿真语言也多用高级描述语言，如C/C++、SystemVerilog、SystemC等。

当确定了系统的整体功能后，系统工程师就要对整个系统进行功能划分，定义接口及其主要模块(如时钟、协议等)，并且通过仿真比较不同的方案所达到的性能优劣，根据市场和系统性能指标要求从整体上优化实现方案，以满足设计要求。

2. 寄存器传输级描述

当系统功能模块和接口定义好以后，逻辑工程师就开始规划描述寄存器到寄存器之间的逻辑功能实现。寄存器传输级描述不同于门级描述，它比门级更加抽象，它不关心寄存器与组合逻辑之间的细节关系，实现的语言也比门级简单。