

计 算 机 科 学 丛 书

HZ BOOKS  
华章教育

ELSEVIER  
爱思唯尔

原书第2版

# 数字设计和 计算机体系结构

[美] 戴维·莫尼·哈里斯 (David Money Harris) 莎拉 L. 哈里斯 (Sarah L. Harris) 著

哈维玛德学院

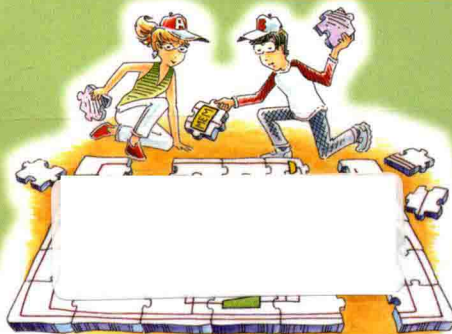
陈俊颖

译  
华南理工大学

Digital Design and Computer Architecture  
Second Edition

Digital Design and  
Computer Architecture

SECOND EDITION



David Money Harris & Sarah L. Harris

MK  
Morgan Kaufmann



机械工业出版社  
China Machine Press

计 算 机 科 学 丛 书

原书第2版

# 数字设计和 计算机体系结构

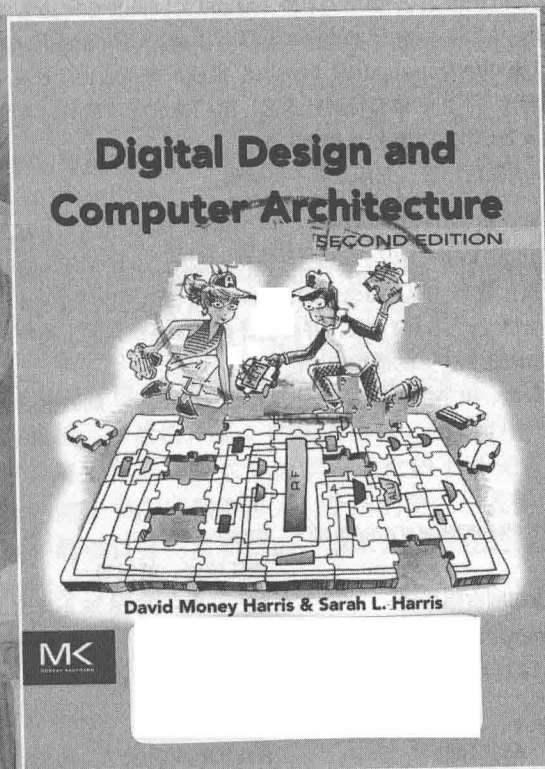
[美] 戴维·莫尼·哈里斯 (David Money Harris) 莎拉 L. 哈里斯 (Sarah L. Harris) 著

哈佛玛德学院

陈俊颖 译

华南理工大学

Digital Design and Computer Architecture  
Second Edition



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

数字设计和计算机体系结构 (原书第 2 版) / (美) 哈里斯 (Harris, D. M.) 等著; 陈俊颖译.  
—北京: 机械工业出版社, 2016.4

(计算机科学丛书)

书名原文: Digital Design and Computer Architecture, Second Edition

ISBN 978-7-111-53451-8

I. 数… II. ①哈… ②陈… III. ①数字电路-电路设计 ②计算机体系结构  
IV. ①TN79 ②TP303

中国版本图书馆 CIP 数据核字 (2016) 第 063290 号

**本书版权登记号: 图字: 01-2012-7899**

Digital Design and Computer Architecture, Second Edition

David Money Harris and Sarah L.Harris

ISBN:978-0-12-394424-5

Copyright © 2013 by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2016 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Printed in China by China Machine Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR, Macau SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权机械工业出版社在中国大陆境内独家出版和发行。本版仅限在中国境内 (不包括香港特别行政区、澳门特别行政区及台湾地区) 出版及标价销售。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

本书以一种流行的方式介绍了从计算机组成和设计到更细节层次的内容, 涵盖了数字逻辑设计的主要内容, 展示了使用 VHDL 和 Verilog 这两种主要硬件描述语言设计 MIPS 处理器的技术细节, 并通过 MIPS 微处理器的设计强化数字逻辑的概念。本书的典型特色是将数字逻辑和计算机体系结构融合, 教学内容反映了当前数字电路设计的主流方法, 并突出计算机体系结构的工程特点, 书中的大量示例及习题也可以加强读者对基本概念和技术的理解和记忆。

本书不仅适合作为计算机、电子工程、电气与控制等专业的教材, 同时也适合从事数字电路设计的工程师和技术人员参考。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 盛思源

责任校对: 董纪丽

印刷: 北京市荣盛彩色印刷有限公司

版次: 2016 年 4 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 31

书号: ISBN 978-7-111-53451-8

定价: 89.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：[www.hzbook.com](http://www.hzbook.com)

电子邮件：[hzjsj@hzbook.com](mailto:hzjsj@hzbook.com)

联系电话：(010)88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

数字逻辑设计、计算机体系结构、嵌入式系统和片上系统设计等课程是计算机系统课程的主体。本书巧妙地将数字设计和计算机体系结构融合在一起，既明确了数字设计是计算机体系结构的基础知识，也让读者了解了计算机体系结构课程如何运用数字设计课程中的关键知识。本书各章节知识连贯衔接，自然而然地引导读者从最基本的0和1一直深入到计算机微处理器的构建。通过本书，完全没有计算机系统和软硬件知识的学生，也能从零开始循序渐进地掌握设计计算机微处理器以及编写相应程序的基本原理和方法。

“层次化、模块化、规整化”三大计算机软硬件通用的设计原则，贯穿本书始终。通过这样的设计思想学习，读者能建立良好的工程设计思路，为将来设计大规模的复杂软硬件系统打下良好的基础。同时，第2版的内容紧密贴近领域新动态，书中涉及的相关数据、编程语言、软件工具、硬件结构等都紧跟行业发展。在征得原书作者和原出版社同意的情况下，本书增加了附录D“MIPS处理器的FPGA实现”，补充在实际开发板和软件开发环境上设计和实现计算机微处理器系统的相关内容。通过本书的学习，能增强读者使用主流工具和开发环境进行实际应用设计的能力。

此外，本书内容丰富充实，文字通俗流畅，叙述风趣幽默，并配有大量示例和习题，有助于读者理解和掌握数字设计和计算机体系结构的相关知识。本书不仅适合用于相关专业课程的教学，也适合作为相关工程技术人员的参考书籍。

本书由华南理工大学陈俊颖翻译定稿。在本书的完成过程中，华南理工大学的陈虎(原书第1版译者)和闵华清等老师给予了大力的支持与帮助，机械工业出版社的姚蕾等编辑提出了宝贵的意见并付出了辛勤的劳动，Imagination Technologies公司的RobertOwen、Laurence Keung和工程师提供了积极的建议和技术支持，在此对他们表示衷心的感谢!

在本书翻译过程中，译者力求准确无误地表达原文意思，尽可能使文字流畅易懂。但是受译者水平和时间所限，难免有疏漏和错误之处，恳请广大读者不吝指正。

最后，特别感谢我的家人一直以来对我无私的关爱。

译者

2016年1月6日

本书以一种易于接受的方式介绍了从计算机组成和设计到更细节层次的教学内容，展现了如何使用 VHDL 和 System Verilog 语言设计 MIPS 处理器的技术细节。为学生提供在现代 FPGA 上实现大型数字系统设计的机会。书中提供的方法既向学生传授了知识又具有启发性。

——David A. Patterson, 加利福尼亚大学伯克利分校

本书为传统的教学内容提供了新的视角。很多教科书看上去像繁杂的灌木丛，作者在本书中将“枯枝”去除，同时保留了最基本的内容，并把这些内容放到了现代的环境中。因此，他们提供的教材可以激发学生未来挑战设计方案的兴趣。

——Jim Frenzel, 爱达荷大学

Harris 的写作风格引人入胜，而且能提供很多知识。他们对材料的运用水平很高，通过大量的图来引导学生进入计算机工程领域。组合逻辑电路、微体系结构和存储器系统等内容处理得非常好。

——James Pinter-Lucke, 克莱蒙麦肯纳学院

Harris 的这本书非常清晰且易于理解。习题的设计非常好，同时也提供了很多现实案例。这本书避免了许多其他教材中冗长而费解的解释。很明显，作者花费了很多时间和努力提高这本书的可读性。本人强烈推荐这本书。

——Peiyi Zhao, 查普曼大学

Harris 撰写了一部成功融合数字系统设计和计算机体系结构的教材。这是一本很受欢迎的教科书，它介绍了很多数字系统设计的内容，同时详细解释了 MIPS 体系结构的细节。本人强烈推荐这本书。

——James E. Stine, Jr., 俄克拉荷马州立大学

这是一本令人印象深刻的书。Harris 将晶体管、电路、逻辑门、有限状态机、存储器、算术部件等微处理器设计中的所有重要元素完美地结合在一起，并最终引出计算机体系结构。这本书为理解如何完美地设计复杂系统提供了很好的指导。

——Jaeha Kim, Rambus 公司

这是一本写得非常好的书，不仅适用于第一次学习这些领域的年轻工程师，而且可以为有经验的工程师提供参考。本人强烈推荐这本书。

——A. Utku Diril, Nvidia 公司

目前已经有很多优秀的数字逻辑设计书籍，也有一些很好的计算机体系结构教材（例如，Patterson 和 Hennessy 撰写的经典教材），为什么还需要出版一本包含了数字逻辑设计和体系结构的书呢？本书的独特之处在于从计算机体系结构的视角来学习数字逻辑设计，内容从基本的二进制开始，直到引导学生完成 MIPS 处理器的设计。

多年来，我们曾在哈佛学院使用了多个版本的《Computer Organization and Design》（计算机组成与设计）（由 Patterson 和 Hennessy 撰写）。我们特别欣赏该书覆盖了 MIPS 处理器的体系结构和微体系结构，因为 MIPS 处理器是获得商业成功的体系结构，而且它也非常简单，可以在导论课程中向学生解释清楚，并可以由学生自主设计和实现。由于我们的课程没有预修课程，所以前半个学期需要介绍数字逻辑设计，而这部分没有被《Computer Organization and Design》所包含。其他大学也表示需要一本能包含数字电路设计和体系结构的教材。于是，我们着手开始准备这样一本包含了数字逻辑设计和体系结构的书。

我们相信设计处理器对于电子工程和计算机专业的学生是一个特殊而重要的经历。对外行而言，处理器内部的工作几乎像魔术一样，然而事实证明，如果详细解释，处理器的工作原理就非常易于理解。数字逻辑设计本身是一个令人激动的主题。汇编语言程序则揭示了处理器内部所用的语言。而微体系结构将两者联系在一起。

本书适合于在一个学期内完成教学的数字逻辑设计和计算体系结构入门课程，也可以用于两个学期的教学，以使用更多的时间来消化和理解书中所讲的知识并在实验室中进行实践。不需要任何预修内容也可以教授这个课程。本书一般在大学本科二年级或者三年级使用，也可以提供给聪明的一年级学生学习。

## 特点

本书有以下特点。

### 并列讲述 SystemVerilog 和 VHDL 语言

硬件描述语言（Hardware Description Language, HDL）是现代数字逻辑设计实践的中心，而设计者分成了 SystemVerilog 语言和 VHDL 语言两个阵营。在介绍了组合逻辑和时序逻辑设计后，本书紧接着就在第 4 章介绍硬件描述语言。硬件描述语言将在第 5 章和第 7 章用于设计更大的模块和整个处理器。然而，如果不讲授硬件描述语言，那么可以跳过第 4 章，而后续章节仍然可以继续使用。

本书的特色在于使用并列方式讲述 SystemVerilog 和 VHDL，使读者可以快速地对两种语言。第 4 章描述适用于这两种硬件描述语言的原则，而且并列给出了这两种语言的语法和实例。这种并列方法使得在教学中教师可以选择其中一种硬件描述语言来讲述，也可以让读者在专业实践中很快地从一种描述语言转到另一种描述语言。

### 经典的 MIPS 体系结构和微体系结构

第 6 章和第 7 章主要介绍 MIPS 体系结构。这部分内容主要改编自 Patterson 和 Hennessy 的论著。MIPS 是一个理想的体系结构，因为每年有上百万实际产品投入使用，而且高效和易于学习。同时，世界各地上百所大学已经围绕 MIPS 体系结构开发了教学内容、实验和工具。

## 现实视角

第6、7、8章列举了Intel公司x86处理器系列的体系结构、微体系结构和存储器层次结构。第8章还介绍了Microchip PIC32微控制器的外部设备。这些章节揭示了书中所讲的概念如何应用到很多PC内部芯片和消费电子产品的设计中。

## 高级微体系结构概览

第7章介绍了现代高性能微体系结构的特征,包括分支预测、超标量、乱序执行操作、多线程和多核处理器。这些内容对于第一次上体系结构课程的学生比较容易理解,并说明了本书介绍的微体系结构原理是如何扩展到现代处理器的设计中的。

## 章末的习题和面试问题

学习数字设计的最佳方式是实践。每章的最后有很多习题来实际应用所讲述的内容。习题后面是同行向申请这个领域工作的学生提出的一些面试问题。这些问题可以让学生感受到面试过程中可能遇见的典型问题。习题答案可以通过本书的配套网站和教师网站获得。更详细的内容参见下文——在线补充资料。

## 在线补充资料<sup>⊖</sup>

补充材料可以通过 [textbooks.elsevier.com/9780123944245](http://textbooks.elsevier.com/9780123944245) 获得。本书配套网站(对所有读者开放)包括了以下内容:

- 奇数编号习题的答案。
- Altera® 和 Synopsys® 公司专业版计算机辅助设计工具的链接。
- QtSpim(一般称为 SPIM)的链接,一个 MIPS 模拟器。
- MIPS 处理器的硬件描述语言(HDL)代码。
- Altera Quartus II 工具的提示。
- Microchip MPLAB IDE(集成开发环境)工具的提示。
- PPT 格式的电子教案。
- 课程示例和实验素材。
- 勘误表。

教师网站(链接到本书配套网站,仅提供给在 [textbooks.elsevier.com](http://textbooks.elsevier.com) 注册的使用者)包括:

- 所有习题的答案。
- Altera® 和 Synopsys® 公司专业版计算机辅助设计工具的链接(Synopsys 公司为取得资格认证的大学提供 Synplify® Premier 工具的 50 个许可证。更多 Synopsys 大学计划内容请参见本书教师网站)。
- JPG 格式和 PPT 格式的书中插图。

关于在课程中使用 Altera、Synopsys、Microchip 和 QtSpim 工具的更详细的内容请参见下文。构建实验工具的细节也将在下面介绍。

⊖ 关于本书教辅资源,使用教材的教师需通过爱思唯尔的教材网站([www.textbooks.elsevier.com](http://www.textbooks.elsevier.com))注册并通过审批后才能获取。具体方法如下:在 [www.textbooks.elsevier.com](http://www.textbooks.elsevier.com) 教材网站查找到该书后,点击“instructor manual”便可申请查看该教师手册。有任何问题,请致电 010-85208853。——编辑注



## 如何使用课程中的软件工具

### Altera Quartus II

Quartus II Web Edition 是 Quartus™ II FPGA 设计工具的免费版本。基于此软件，学生可以使用原理图或者硬件描述语言(SystemVerilog 或 VHDL)完成数字逻辑设计。在完成设计后，学生可以使用 Altera Quartus II Web Edition 中包含的 ModelSim™-Altera Starter Edition 工具模拟电路。Quartus II Web Edition 还包含支持 SystemVerilog 或者 VHDL 的内置逻辑综合工具。

Web Edition 和 Subscription Edition 两个软件的区别在于，Web Edition 仅支持 Altera 公司部分常用的 FPGA 器件。ModelSim-Altera Starter Edition 和 ModelSim 商业版的区别在于，Starter Edition 降低了 10 000 多行硬件描述语言代码的模拟速度。

### Microchip MPLAB IDE

Microchip MPLAB IDE 是用于 PIC 微控制器编程的工具，可免费下载。MPLAB 将程序的编写、编译、模拟和调试集成到一个界面。它包括一个 C 编译器和调试器，允许学生开发 C 语言和汇编程序，编译它们，以及可选择地将它们编程到 PIC 微控制器。

### 可选工具：Synplify Premier 和 QtSpim

Synplify Premier 和 QtSpim 是本课程资料的可选工具。

Synplify Premier 产品是一个面向 FPGA 和 CPLD 设计的综合和调试环境。它包含 HDL Analyst，一个独特的图形化 HDL 分析工具，自动生成可以回 HDL 源代码交叉探测的设计示意图。在学习和调试过程中这非常有用。

Synopsys 公司为取得资格认证的大学提供 Synplify® Premier 工具的 50 个许可证。更多关于 Synopsys 大学计划内容或者 Synopsys FPGA 设计软件信息，请参见本书教师网站 ([textbooks.elsevier.com/9780123944245](http://textbooks.elsevier.com/9780123944245))。

QtSpim(简称为 SPIM)是一个可运行 MIPS 汇编代码的 MIPS 模拟器。学生可以在文本文件中输入 MIPS 汇编代码，通过 QtSpim 进行模拟。QtSpim 显示指令、存储器和寄存器的值。用户手册和示例文件的链接可以通过本书配套网站 ([textbooks.elsevier.com/9780123944245](http://textbooks.elsevier.com/9780123944245)) 访问。

## 实验

配套网站提供了从数字逻辑设计到计算机体系结构的一系列实验的链接。这些实验教学生如何使用 Quartus II 工具来输入、模拟、综合和实现他们的设计。这些实验也包含了使用 Microchip MPLAB IDE 完成 C 语言和汇编语言编程的内容。

经过综合后，学生可以在 Altera DE2 开发和教育板上实现自己的设计。这个功能强大且具有价格优势的开发板可以通过 [www.altera.com](http://www.altera.com) 获得。该开发板包含可通过编程来实现学生设计的 FPGA。我们提供的实验描述了如何使用 Cyclone II Web Edition 在 DE2 开发板上实现一些设计。

为了运行这些实验，学生需要下载并安装 Altera Quartus II Web Edition 和 Microchip MPLAB IDE。教师也需要选择软件安装在实验室的机器上。这些实验包括如何在 DE2 开发板上实现项目的指导。这些实现步骤可以跳过，但是我们认为它有很大的价值。

我们在 Windows 平台上测试了所有的实验，当然这些工具也可以在 Linux 上使用。

## 错误

正如所有有经验的程序员所知道的，比较复杂的程序都毫无疑问存在潜在的错误。本书也不例外。我们花费了大量的精力查找和去除本书的错误。然而，错误仍然不可避免。我们将在本书的网站上维护和更新勘误表。

请将你发现的错误发送到 [ddcabugs@onehotlogic.com](mailto:ddcabugs@onehotlogic.com)。第一个报告实质性错误而且在后续版本中采用了其修改意见的读者可以得到 1 美元的奖励！

## 致谢

首先，我们要感谢 David Patterson 和 John Hennessy。他们在《Computer Organization and Design》(计算机组成与设计)一书中对 MIPS 微体系结构进行了开创性的介绍。我们多年以来讲授了该书的多个版本。感谢他们对这本书的慷慨支持，以及允许在他们的微体系结构上进行设计。

我们喜爱的卡通画家 Duane Bibby 花费了很长时间和努力来说明数字电路设计中有趣的奇遇。我们也很感激 Morgan Kaufmann 公司的 Nate McFadden、Todd Green、Danielle Miller、Robyn Day 以及团队的其他同事，没有他们的热情支持，本书将无法面世。

我们要感谢 Matthew Watkins 为第 7 章“异构多处理器”一节撰稿。我们也感谢 Chris Parks、Carl Pearson 和 Johnathan Chai 为本书第 2 版测试代码和开发内容。

很多评阅人也对本书的质量给予了很大的帮助。他们包括：John Barr、Jack V. Briner、Andrew C. Brown、Carl Baumgaertner、A. Utku Diril、Jim Frenzel、Jaeha Kim、Phillip King、James Pinter-Lucke、Amir Roth、Z. Jerry Shi、James E. Stine、Luke Teysier、Peiyi Zhao、Zach Dodds、Nathaniel Guy、Aswin Krishna、Volnei Pedroni、Karl Wang、Ricardo Jasinski 以及一位匿名评阅人。

我们也非常感谢哈佛玛德学院上这个课程的学生，他们对本书的草稿提供了有帮助的反馈。需要特别记住的是：Matt Weiner、Carl Walsh、Andrew Carter、Casey Schilling、Alice Clifton、Chris Acon 和 Stephen Brawner。

最后，但同样重要的是，我们要感谢家人的爱和支持。

# 目 录

Digital Design and Computer Architecture, Second Edition

出版者的话	1.7.5 CMOS 非门	19
译者序	1.7.6 其他 CMOS 逻辑门	19
本书赞誉	1.7.7 传输门	20
前言	1.7.8 类 nMOS 逻辑	20
<b>第 1 章 二进制</b>	1.8 功耗*	21
1.1 课程计划	1.9 总结和展望	22
1.2 控制复杂性的艺术	习题	22
1.2.1 抽象	面试问题	28
1.2.2 约束	<b>第 2 章 组合逻辑设计</b>	29
1.2.3 三 Y 原则	2.1 引言	29
1.3 数字抽象	2.2 布尔表达式	31
1.4 数制	2.2.1 术语	31
1.4.1 十进制数	2.2.2 与或式	31
1.4.2 二进制数	2.2.3 或与式	32
1.4.3 十六进制数	2.3 布尔代数	33
1.4.4 字节、半字节和全字	2.3.1 公理	33
1.4.5 二进制加法	2.3.2 单变量定理	33
1.4.6 有符号的二进制数	2.3.3 多变量定理	34
1.5 逻辑门	2.3.4 定理的统一证明方法	35
1.5.1 非门	2.3.5 等式化简	36
1.5.2 缓冲器	2.4 从逻辑到门	37
1.5.3 与门	2.5 多级组合逻辑	39
1.5.4 或门	2.5.1 减少硬件	39
1.5.5 其他两输入逻辑门	2.5.2 推气泡	40
1.5.6 多输入门	2.6 X 和 Z	41
1.6 数字抽象	2.6.1 非法值 X	41
1.6.1 电源电压	2.6.2 浮空值 Z	41
1.6.2 逻辑电平	2.7 卡诺图	42
1.6.3 噪声容限	2.7.1 画圈的原理	43
1.6.4 直流电压传输特性	2.7.2 卡诺图化简逻辑	44
1.6.5 静态约束	2.7.3 无关项	46
1.7 CMOS 晶体管*	2.7.4 小结	47
1.7.1 半导体	2.8 组合逻辑模块	47
1.7.2 二极管	2.8.1 复用器	47
1.7.3 电容	2.8.2 译码器	50
1.7.4 nMOS 和 pMOS 晶体管	2.9 时序	51

2.9.1 传播延迟和最小延迟 .....	51	第4章 硬件描述语言 .....	101
2.9.2 毛刺 .....	54	4.1 引言 .....	101
2.10 总结 .....	55	4.1.1 模块 .....	101
习题 .....	55	4.1.2 硬件描述语言的起源 .....	101
面试问题 .....	60	4.1.3 模拟和综合 .....	102
<b>第3章 时序逻辑设计</b> .....	<b>61</b>	4.2 组合逻辑 .....	104
3.1 引言 .....	61	4.2.1 位运算符 .....	104
3.2 锁存器和触发器 .....	61	4.2.2 注释和空白 .....	104
3.2.1 SR 锁存器 .....	62	4.2.3 缩位运算符 .....	106
3.2.2 D 锁存器 .....	63	4.2.4 条件赋值 .....	106
3.2.3 D 触发器 .....	64	4.2.5 内部变量 .....	106
3.2.4 寄存器 .....	64	4.2.6 优先级 .....	109
3.2.5 带使能端的触发器 .....	65	4.2.7 数字 .....	109
3.2.6 带复位功能的触发器 .....	65	4.2.8 Z 和 X .....	110
3.2.7 晶体管级锁存器和 触发器的设计* .....	66	4.2.9 位混合 .....	111
3.2.8 小结 .....	66	4.2.10 延迟 .....	112
3.3 同步逻辑设计 .....	67	4.3 结构化建模 .....	113
3.3.1 一些有问题的电路 .....	67	4.4 时序逻辑 .....	115
3.3.2 同步时序电路 .....	68	4.4.1 寄存器 .....	115
3.3.3 同步电路和异步电路 .....	70	4.4.2 复位寄存器 .....	115
3.4 有限状态机 .....	70	4.4.3 带使能端的寄存器 .....	116
3.4.1 有限状态机设计实例 .....	70	4.4.4 多寄存器 .....	116
3.4.2 状态编码 .....	74	4.4.5 锁存器 .....	119
3.4.3 Moore 型状态机和 Mealy 型状态机 .....	76	4.5 更多组合逻辑 .....	119
3.4.4 状态机的分解 .....	79	4.5.1 case 语句 .....	120
3.4.5 由电路图导出状态机 .....	80	4.5.2 if 语句 .....	122
3.4.6 有限状态机小结 .....	82	4.5.3 带有无关项的真值表 .....	122
3.5 时序逻辑的时序 .....	82	4.5.4 阻塞赋值和非阻塞赋值 .....	124
3.5.1 动态约束 .....	83	4.6 有限状态机 .....	127
3.5.2 系统时序 .....	83	4.7 数据类型* .....	130
3.5.3 时钟偏移* .....	87	4.7.1 SystemVerilog .....	130
3.5.4 亚稳态 .....	88	4.7.2 VHDL .....	131
3.5.5 同步器 .....	89	4.8 参数化模块* .....	133
3.5.6 分辨时间的推导* .....	90	4.9 测试程序 .....	135
3.6 并行 .....	92	4.10 总结 .....	138
3.7 总结 .....	94	习题 .....	138
习题 .....	95	面试问题 .....	145
面试问题 .....	100	<b>第5章 数字模块</b> .....	<b>146</b>
		5.1 引言 .....	146
		5.2 算术电路 .....	146

5.2.1	加法	146	6.3.4	解释机器语言代码	189
5.2.2	减法	151	6.3.5	存储程序	189
5.2.3	比较器	152	6.4	编程	190
5.2.4	算术逻辑单元	153	6.4.1	算术/逻辑指令	190
5.2.5	移位器和循环移位器	154	6.4.2	分支	193
5.2.6	乘法*	155	6.4.3	条件语句	194
5.2.7	除法*	156	6.4.4	循环	195
5.2.8	补充阅读	156	6.4.5	数组	197
5.3	数制	157	6.4.6	函数调用	200
5.3.1	定点数	157	6.5	寻址方式	206
5.3.2	浮点数*	157	6.6	编译、汇编和装入	208
5.4	时序电路模块	160	6.6.1	内存映射	208
5.4.1	计数器	160	6.6.2	转换成二进制代码和 开始执行程序	209
5.4.2	移位寄存器	160	6.7	其他主题*	211
5.5	存储器阵列	162	6.7.1	伪指令	211
5.5.1	概述	162	6.7.2	异常	212
5.5.2	动态随机访问存储器	164	6.7.3	有符号指令和无符号指令	212
5.5.3	静态随机访问存储器	165	6.7.4	浮点指令	213
5.5.4	面积和延迟	165	6.8	从现实世界看: x86 结构*	214
5.5.5	寄存器文件	165	6.8.1	x86 寄存器	215
5.5.6	只读存储器	165	6.8.2	x86 操作数	215
5.5.7	使用存储器阵列的逻辑	167	6.8.3	状态标志	216
5.5.8	存储器 HDL	167	6.8.4	x86 指令集	216
5.6	逻辑阵列	168	6.8.5	x86 指令编码	218
5.6.1	可编程逻辑阵列	169	6.8.6	x86 的其他特性	219
5.6.2	现场可编程逻辑门阵列	169	6.8.7	小结	219
5.6.3	阵列实现*	173	6.9	总结	219
5.7	总结	174	习题	220	
习题	174	面试问题	227		
面试问题	180				
<b>第 6 章 体系结构</b>	<b>181</b>	<b>第 7 章 微体系结构</b>	<b>228</b>		
6.1	引言	181	7.1	引言	228
6.2	汇编语言	182	7.1.1	体系结构状态和指令集	228
6.2.1	指令	182	7.1.2	设计过程	229
6.2.2	操作数: 寄存器、存储器和 常数	183	7.1.3	MIPS 微体系结构	230
6.3	机器语言	187	7.2	性能分析	230
6.3.1	R 类型指令	187	7.3	单周期处理器	231
6.3.2	I 类型指令	188	7.3.1	单周期数据路径	231
6.3.3	J 类型指令	189	7.3.2	单周期控制	235
			7.3.3	更多指令	237

7.3.4 性能分析 .....	239	8.3.5 MIPS 高速缓存的发展* .....	314
7.4 多周期处理器 .....	240	8.4 虚拟存储器 .....	315
7.4.1 多周期数据路径 .....	240	8.4.1 地址转换 .....	316
7.4.2 多周期控制 .....	244	8.4.2 页表 .....	317
7.4.3 更多指令 .....	252	8.4.3 转换后备缓冲器 .....	318
7.4.4 性能分析 .....	254	8.4.4 存储器保护 .....	319
7.5 流水线处理器 .....	255	8.4.5 替换策略* .....	319
7.5.1 流水线数据路径 .....	257	8.4.6 多级页表* .....	319
7.5.2 流水线控制 .....	258	8.5 I/O 简介 .....	321
7.5.3 冲突 .....	258	8.6 嵌入式 I/O 系统 .....	322
7.5.4 更多指令 .....	269	8.6.1 PIC32MX675F512H 微控制器 .....	322
7.5.5 性能分析 .....	269	8.6.2 通用数字 I/O .....	326
7.6 硬件描述语言表示* .....	270	8.6.3 串行 I/O .....	328
7.6.1 单周期处理器 .....	270	8.6.4 计时器 .....	336
7.6.2 通用模块 .....	274	8.6.5 中断 .....	337
7.6.3 基准测试程序 .....	276	8.6.6 模拟 I/O .....	339
7.7 异常* .....	279	8.6.7 其他微控制器外设 .....	343
7.8 高级微体系结构* .....	281	8.7 PC I/O 系统 .....	358
7.8.1 深流水线 .....	282	8.7.1 USB .....	359
7.8.2 分支预测 .....	282	8.7.2 PCI 和 PCI Express .....	360
7.8.3 超标量处理器 .....	284	8.7.3 DDR3 内存 .....	360
7.8.4 乱序处理器 .....	285	8.7.4 网络 .....	360
7.8.5 寄存器重命名 .....	287	8.7.5 SATA .....	361
7.8.6 单指令流多数据 .....	288	8.7.6 连接到 PC .....	361
7.8.7 多线程 .....	288	8.8 从现实世界看: x86 存储器和 I/O 系统* .....	363
7.8.8 同构多处理器 .....	289	8.8.1 x86 高速缓存系统 .....	363
7.8.9 异构多处理器 .....	289	8.8.2 x86 虚拟存储器 .....	364
7.9 从现实世界看: x86 微体系 结构* .....	290	8.8.3 x86 可编程 I/O .....	365
7.10 总结 .....	295	8.9 总结 .....	365
习题 .....	296	后记 .....	365
面试问题 .....	300	习题 .....	365
		面试问题 .....	371
<b>第 8 章 存储器和输入/输出系统</b> ...	<b>301</b>		
8.1 引言 .....	301	附录 A 数字系统实现 .....	372
8.2 存储器系统性能分析 .....	304	附录 B MIPS 指令 .....	396
8.3 高速缓存 .....	305	附录 C C 语言编程 .....	400
8.3.1 高速缓存中存放的数据 .....	305	附录 D MIPS 处理器的 FPGA 实现 .....	433
8.3.2 高速缓存中的数据查找 .....	305	补充阅读 .....	452
8.3.3 数据的替换 .....	311	索引 .....	454
8.3.4 高级高速缓存设计* .....	312		

## 二 进 制

## 1.1 课程计划

在过去的三十年里，微处理器彻底变革了我们的世界。现在一台笔记本电脑的计算能力都远远超过了过去一个房间大小的计算机。高级汽车上包含了大约 50 个微处理器。微处理器的进步使得移动电话和因特网(Internet)成为可能，极大地促进了医学的进步，也改变了战争的式样。全球集成电路工业销售额从 1985 年的 210 亿美元发展到 2012 年的 3000 亿美元，其中微处理器占到了重要部分。我们相信微处理器不仅仅只是对技术、经济和社会有重要意义，而且它也潜在地激发了人类的创造力。在读者学习完本书后，将学会如何设计和构造一个属于自己的微处理器。这些基本技能将为读者设计其他数字系统奠定坚实的基础。

我们假设读者对电子学有基本的认识，有一定的编程经验和基础，同时对理解微处理器的内部运行原理有真正的兴趣。本书将集中讨论基于 0 和 1 的数字系统的设计。我们从接收 0 和 1 作为输入，产生 0 和 1 作为输出的逻辑门开始本课程。接着，我们将研究如何利用这些逻辑门构成加法器、存储器等比较复杂的模块。随后，我们将学习使用微处理器的语言(汇编语言)进行程序设计。最后，我们将上述内容结合起来构造一个能执行汇编程序的微处理器。

数字系统的一个重要特点是其构造模块相当简单：仅仅包括 0 和 1。它不需要繁杂的数学知识或高深的物理学知识。相反，设计者的最大挑战是如何将这些简单的部件组合起来构成复杂的系统。微处理器可能是读者构造的第一个复杂系统，其复杂性可能一下子难以全部接受。因此，如何控制复杂性是贯穿全书的一个重要主题。

## 1.2 控制复杂性的艺术

与非专业人员相比，计算机科学家或工程师的一个重要特征是掌握了系统地控制复杂性的方法。现代数字系统由上百万，甚至数十亿的晶体管构成。没有人能通过为每个晶体管的电子运动建立并求解方程的方法来理解这样的系统。读者必须学会如何控制复杂性，从而理解如何在陷入细节的情况下构造微处理器系统。

## 1.2.1 抽象

管理复杂性的关键技术在于抽象(abstraction)：隐蔽不重要的细节。一个系统可以从多个不同层面抽象。例如，美国的政治家将世界抽象为城市、县、州和国家。一个县包含了多个城市，而一个州则包含了多个县。当一个政治家竞选总统时，他更对整个州的投票情况有兴趣，而不是单个县。因此，州在这个层次中的抽象更有益处。另一方面，美国人口调查局需要统计



每个城市的人口，必须考虑更低层次抽象的细节。

图 1-1 给出了一个电子计算机系统的抽象层次，其中在每个层次中都包含了典型的模块。最底层的抽象是物理层，即电子的运动。电子的行为由量子力学和麦克斯韦 (Maxwell) 方程描述。系统由晶体管或以前的真空管等电子器件 (device) 构成。这些器件都有明确定义的称为称端子 (terminal) 的外部连接点，并建立了每个端子上的电压和电流之间的关系模型。通过器件级的抽象，我们可以忽略单个电子。更高一级的抽象为模拟电路 (analogy circuit)。在这一级中，器件组合在一起构造成放大器等组件。模拟电路的输入和输出都是连续的电压值。逻辑门等数字电路 (digital circuit) 则将电压控制在离散的范围，以便表示 0 和 1。在逻辑设计中，我们将使用数字电路构造更复杂的结构，例如加法器或存储器。

微结构将逻辑和体系结构层次的抽象连接在一起。体系结构 (architecture) 层描述了程序员观点的计算机抽象。例如，目前广泛应用于个人计算机 (Personal Computer, PC) 的 Intel x86 体系结构定义了一套指令系统和寄存器 (用于存储临时变量的存储器)，程序员可以使用这些指令和寄存器。微结构将逻辑元素组合在一起来实现体系结构中定义的指令。一个特定的体系结构可以有不同的微结构实现方式，以便取得在价格、性能和功耗等方面的不同折中。例如，Intel Core i7、Intel 80486 和 AMD Athlon 等都是 x86 体系结构的三种不同的微结构实现。

进入软件层面后，操作系统负责处理底层的抽象，例如访问硬盘或管理存储器。最后，应用软件使用操作系统提供的这些功能来解决用户的问题。正是借助于抽象的威力，年迈的祖母可以通过计算机上网，而不用考虑电子的量子波动 (vibration) 或计算机中的存储器组织问题。

本书主要讨论从数字电路到体系结构之间的抽象层次。当读者处于某个抽象层次时，最好能了解当前抽象层次之上和之下的层次。例如，计算机科学家不可能在不理解程序运行平台的体系结构的情况下来充分优化代码。在不了解晶体管具体用途的情况下，器件工程师也不能在晶体管设计时做出明智的设计选择。我们希望读者学习完本书后，能选择正确的层次解决问题，同时评估自己的设计选择对其他抽象层次的影响。

## 1.2.2 约束

约束 (discipline) 是对设计选择的一种内在限制，通过这种限制可以更有效地在更高的抽象层次上工作。使用可互换部件是约束的一种常见应用，其典型例子是来复枪 (flintlock rifle) 的制作。在 19 世纪早期，来复枪靠手工一支支地制作。来复枪的零件从很多不同的手工制作商那里买来，然后由一个技术熟练的做枪工人组装在一起。基于可互换部件的约束变革了这个产业。通过将零件限定为一个误差允许范围内的标准集合，就可以很快地组装和修复来复枪，而且不需要太熟练的技术。做枪工人不再需要考虑枪管和枪托形状等较低层次的抽象。

在本书中，对数字电路的约束非常重要。数字电路使用离散电压，而模拟电路使用连续电压。因此，数字电路是模拟电路的子集，而且在某种意义上其能力弱于范围更广的模拟电路。然而，数字电路的设计很简单。通过数字电路的约束规则，我们可以很容易地将组件组合成复

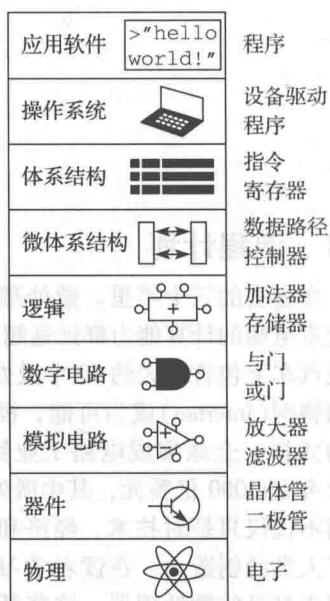


图 1-1 电子计算机系统的抽象层次



杂的系统，而且这种数字系统在很多应用上都远远优于由模拟组件组成的系统。例如，数字化的电视、光盘(CD)以及移动电话正在取代以前的模拟设备。

### 1.2.3 三 Y 原则

除了抽象和约束外，设计者还使用另外三条准则来处理系统的复杂性：层次化(hierarchy)、模块化(modularity)和规整化(regularity)。这些原则对于软硬件的设计都是通用的。

- 层次化：将系统划分为若干模块，然后更进一步划分每个模块直到这些模块可以很容易理解。
- 模块化：所有模块有定义好的功能和接口，以便于它们之间可以很容易地相互连接而不会产生意想不到的副作用。
- 规整化：在模块之间寻求一致，通用模块可以重复使用多次，以便减少设计不同模块的数量。

我们用制作来复枪的例子来解释这三 Y 原则。在 19 世纪早期，来复枪是最复杂的常见物品之一。使用层次化原理，可以将它划分为图 1-2 所示的几个组件：枪机(lock)、枪托和枪管。

枪管是一个长的金属管子，子弹就是通过这里射出的；枪机是一种射击设备；而枪托是用木头制成的，它将各个部分连接起来并为使用者提供牢固的握枪位置。相应地，枪机包含扳机、击锤、燧石、扣簧和药锅。每个组件都可以用进一步层次化来描述。

模块化使得每个组件都应有明确定义的功能和接口。枪托的功能是装配枪机和枪管，它的接口包括长度和装配钉的位置。在模块化的来复枪设计中，如果长度和装配钉的位置都合适，那么来自于不同制造商的枪托可以用于特定的枪管。枪管的功能是使子弹更加精确地射出，模块化设计规定它不能产生副作用：对枪管的设计不能影响枪托的功能。

规整化表明可互换部件是一个好方法。利用规整化原理，一个损坏的枪管可以用相同的部分取代。可以在装配线上更有效地生产枪管，而不是辛苦地手工制作。

层次化、模块化和规整化这三条原则在本书中很重要，它们将贯穿本书的内容。

## 1.3 数字抽象

大部分物理变量是连续的。例如，电线上的电压、震动的频率、物体的位置等都是连续值。相反，数字系统使用离散值变量(discrete-valued variable)来表示信息——也就是说，变量是有限数目的不同离散值。

早期 Charles Babbage 的分析机 (Analytical Engine) 使用具有 10 个离散值变量的数字系统。从 1834 年到 1871 年，Babbage 一直在设计和尝试制作这种机械计算机。分析机使用从 0 号 ~ 9

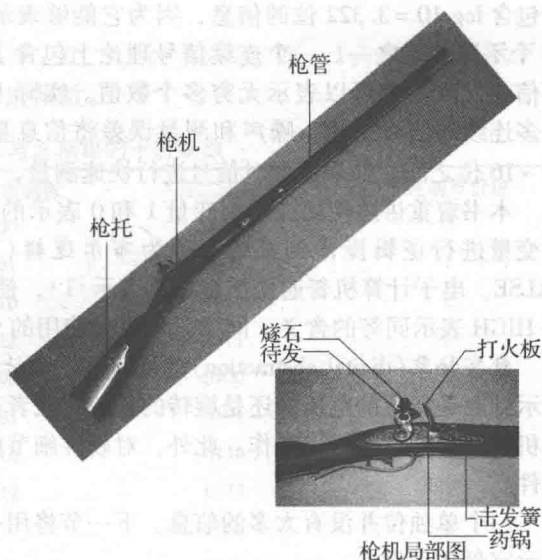


图 1-2 来复枪及其枪机的特写照片

(照片由 Euroarms Italia 提供。www.euroarms.net@2006)