

HZ BOOKS
华章教育



计 算 机 科 学 丛 书

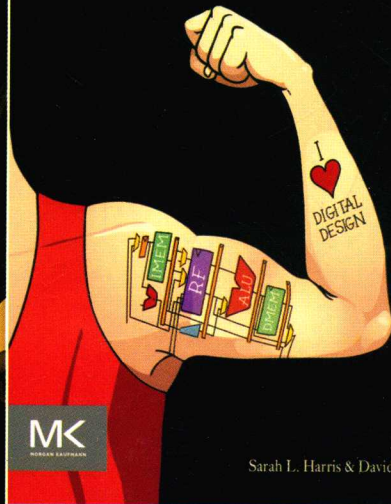
ARM版

数字设计和 计算机体系结构

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

Digital Design and Computer Architecture
ARM Edition

*Digital Design and
Computer Architecture*
ARM® Edition



MK
MOORE KAPLAN

Sarah L. Harris & David Money Harris



机械工业出版社
China Machine Press

计

算

从

书

ARM版

数字设计和 计算机体系结构

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

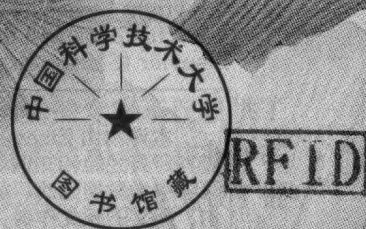
Digital Design and Computer Architecture
ARM Edition

*Digital Design and
Computer Architecture*

ARM® Edition

MK
Morgan Kaufmann

Harris



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·ARM 版) / (美) 莎拉·L·哈里斯 (Sarah L. Harris), (美) 戴维·莫尼·哈里斯 (David Money Harris) 著; 陈俊颖等译. —北京: 机械工业出版社, 2019.6

(计算机科学丛书)

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

ISBN 978-7-111-62925-2

I. 数… II. ①莎… ②戴… ③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 (2019) 第 107386 号

本书版权登记号: 图字 01-2015-6382

ELSEVIER

Elsevier (Singapore) Pte Ltd.

3 Killiney Road, #08-01 Winsland House I, Singapore 239519

Tel: (65) 6349-0200; Fax: (65) 6733-1817

Digital Design and Computer Architecture, ARM Edition

Sarah L. Harris, David Money Harris

Copyright © 2016 Elsevier Inc. All rights reserved.

Materials provided by ARM are copyright © ARM Limited (or its affiliates).

ISBN-13: 9780128000564

This translation of Digital Design and Computer Architecture, ARM Edition by Sarah L. Harris, David Money Harris was undertaken by China Machine Press and is published by arrangement with Elsevier (Singapore) Pte Ltd.

Digital Design and Computer Architecture, ARM Edition by Sarah L. Harris, David Money Harris 由机械工业出版社进行翻译, 并根据机械工业出版社与爱思唯尔(新加坡)私人有限公司的协议约定出版。

《数字设计和计算机体系结构》(原书第 2 版·ARM 版)(陈俊颖 等译)

ISBN: 978-7-111-62925-2

Copyright © 2019 by Elsevier (Singapore) Pte Ltd.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission from Elsevier (Singapore) Pte Ltd. Details on how to seek permission, further information about the Elsevier's permissions policies and arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by Elsevier (Singapore) Pte Ltd. and China Machine Press (other than as may be noted herein).

注意

本译本由 Elsevier (Singapore) Pte Ltd. 和机械工业出版社完成。相关从业及研究人员必须凭借其自身经验和知识对文中描述的信息数据、方法策略、搭配组合、实验操作进行评估和使用。由于医学科学发展迅速, 临床诊断和给药剂量尤其需要经过独立验证。在法律允许的最大范围内, 爱思唯尔、译文的原文作者、原文编辑及原文内容提供者均不对译文或因产品责任、疏忽或其他操作造成的人身及/或财产伤害及/或损失承担责任, 亦不对由于使用文中提到的方法、产品、说明或思想而导致的人身及/或财产伤害及/或损失承担责任。

Printed in China by China Machine Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong SAR, Macau SAR and Taiwan. Unauthorized export of this edition is a violation of the contract.

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

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: 朱秀英

责任校对: 殷虹

印刷: 北京市兆成印刷有限责任公司

版次: 2019 年 7 月第 1 版第 1 次印刷

开本: 185mm×260mm 1/16

印张: 24.25

书号: ISBN 978-7-111-62925-2

定价: 129.00 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 88379833

投稿热线: (010) 88379604

购书热线: (010) 68326294

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中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，不仅擘画了研究的范畴，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，从它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、Afred 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500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

电子邮件：hzsj@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



华章教育

华章科技图书出版中心

赞 誉 |

Digital Design and Computer Architecture, ARM Edition

本书真的太棒了，是全心全意为计算机教学而打造的，并且清楚地表明了两位作者对教学和教育的热爱和热情。读完这本书的学生将在毕业后多年仍感谢他们。写作风格，清晰度，详细图表，信息流程，主题复杂性的逐渐增加，各章节中的精彩例子，章节末尾的习题，简洁明了的解释，有用的现实世界的例子，对每个主题的所有方面的涵盖——所有这些事情都做得很好。如果本书刚好是你的课堂教材，相信你一定会享受这门课，学到真东西，并且在毕业多年后依然受用。

——Mehdi Hatamian, Broadcom 公司高级副总裁

两位作者基于他们的畅销书《Digital Design and Computer Architecture》创造了 ARM 版本，完成得非常出色。重定向到 ARM 是一项具有挑战性的任务，但作者在保持其清晰透彻的演示风格以及出色的文字质量的同时成功完成了这项工作。我相信这个新版本将受到学生和专业人士的欢迎。

——Donald Hung, 圣何塞州立大学

在我作为教授的 10 年间审阅和使用过的所有教科书中，本书是仅有的两本毫无疑问值得购买的教科书（另一本是《Computer Organization and Design》）。它的写作清晰简洁，图表易于理解，用作运行示例的 CPU 足够复杂以在实际中应用，但又足够简单，可以让学生彻底理解。

——Zachary Kurmas, 大谷州立大学

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，作者在这本书中将“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环境中。正因为如此，他们提供的教材可以激发学生为未来的挑战设计解决方案的兴趣。

——Jim Frenzel, 爱达荷大学

两位作者的写作风格令人愉快，内容丰富。他们对资料的处理水平很高，可以通过大量有用的图表向学生介绍计算机工程。他们对组合电路、微结构和存储器系统处理得尤其好。

——James Pinter-Lucke, 克莱蒙特麦肯纳学院

这本书非常清晰而且易于理解。习题的设计非常好，同时也提供了很多现实案例。书中避免了许多其他教材中冗长而令人费解的解释。可以看出，作者花费了很多时间和努力来提高本书的可读性。我强烈推荐这本书。

——Peiyi Zhao, 查普曼大学

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 ARM 体系结构融合在一起，既明确了数字设计作为 ARM 体系结构的基础知识，也帮助读者了解了 ARM 体系结构课程如何运用数字设计课程中的关键知识。本书各章节间衔接连贯，自然而然地引导读者从最基本的 0 和 1 一直深入到 ARM 微处理器的构建。通过这本书，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 ARM 微处理器以及编写相应软硬件程序的基本原理和方法。

“层次化、模块化、规整化”三大计算机软硬件的通用设计原则贯穿本书始终。通过对设计思想的学习，读者能建立起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，本书内容紧密贴近领域新动态，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。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。

本书不仅内容丰富充实，文字通俗流畅，而且叙述风趣幽默，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 ARM 体系结构的相关知识。本书不仅适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。

本书主要由华南理工大学陈俊颖翻译、校对及定稿，付懿轩、庄仁鑫和邢正颖参与了部分翻译工作。本书由广东省自然科学基金项目（编号：2016A030310412）资助完成。在本书翻译过程中，华南理工大学的陈虎（第 1 版译者）给予了大力的支持与帮助，机械工业出版社的曲熠、朱秀英等编辑提出了宝贵的意见并付出了辛勤的劳动，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和支持。

陈俊颖

2019 年 3 月 12 日

本书的独特之处在于从计算机体系结构的角度呈现数字逻辑设计，从 1 和 0 开始，逐步引领读者了解微处理器的设计。

我们相信，构建微处理器是工程和计算机科学专业学生的特殊“仪式”。处理器的内部工作对于不熟悉的人来说似乎是神奇的，但经过仔细解释后，其实是直截了当的。数字设计本身就是一个强大而令人兴奋的主题。汇编语言编程揭示了处理器所使用的内部语言。微体系结构（简称为微结构）将它们链接在一起。

在这本日益流行的图书的前两个版本中，包括了由 Patterson 和 Hennessy 所撰写且被广泛使用的体系结构书籍中讨论的传统 MIPS 体系结构。作为最初的精简指令集计算体系结构之一，MIPS 非常简洁，易于理解和构建。今天，MIPS 仍然是一个重要的体系结构，在 2013 年被 Imagination Technologies 收购后，又被注入了新的活力。

在过去的 20 年中，ARM 体系结构由于其高效和丰富的生态系统而大受欢迎。这段时间出货了超过 500 亿个 ARM 处理器，并且全球超过 75% 的人都在使用带有 ARM 处理器的产品。在撰写本书时，几乎所有在售的手机和平板电脑都包含一个或多个 ARM 处理器。有报道预测数百亿的 ARM 处理器将很快控制物联网。许多公司正在构建高性能 ARM 系统，以在服务器市场挑战 Intel。由于其商业重要性和学生的兴趣，我们撰写了本书的 ARM 版本。

在教学上，MIPS 和 ARM 版本的学习目标是相同的。ARM 体系结构具有许多功能，包括寻址模式和条件执行，这些功能有助于提高效率，但增加了少量的复杂性。它与 MIPS 的微体系结构也非常相似，而条件执行和程序计数器是它们最大的差异。关于 I/O 的章节提供了大量使用 Raspberry Pi 的示例。Raspberry Pi 是一种非常流行的基于 ARM 的嵌入式 Linux 单板计算机。

只要市场依然有需求，我们就希望能够同时提供 MIPS 和 ARM 两个版本。

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。在介绍组合逻辑和时序逻辑设计后，本书紧接着就在第 4 章中介绍硬件描述语言，并将在第 5 章和第 7 章用其来设计处理器的模块和整个处理器。然而，如果不讲授硬件描述语言，第 4 章可以跳过去，不影响后续章节。

本书的特色在于使用并列的方式讲述 SystemVerilog 语言和 VHDL 语言，使得读者可以快速对比两种语言。第 4 章描述了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。这种并列方法使得教师可以选择其中一种硬件描述语言讲述，同时，读者在专业实践中也可以很快从一种描述语言转到另一种描述语言。

ARM 体系结构和微体系结构

第 6 章和第 7 章首次深入介绍了 ARM 体系结构和微体系结构。ARM 是一种理想的体

系结构，因为它是一种每年应用于数百万种产品中的真实体系结构，但又十分精简且易于学习。此外，由于其在商业和业余爱好者世界中的流行，已有多种 ARM 体系结构的模拟和开发工具。在本书中，所有与 ARM 技术相关的材料均经 ARM Limited 许可复制。

现实世界视角

除了讨论 ARM 体系结构的现实世界视角外，第 6 章还介绍了英特尔 x86 处理器的体系结构，以提供另一种视角。第 9 章（在线补充资料[Ⓔ]）还描述了 Raspberry Pi 单板计算机环境中的外围设备，这是一个非常流行的基于 ARM 的平台。这些现实世界视角的章节展示了该章中的概念与许多 PC 和消费电子产品中的芯片之间的关系。

高级微体系结构概览

第 7 章介绍了现代高性能微结构的特征，包括分支预测、超标量、乱序执行、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易于理解，展示了本书中的微结构原理是如何扩展到现代处理器设计中的。

章末的习题和面试问题

学习数字设计的最佳方式是实践。每章末尾都有很多习题用于实践所讲述的内容。习题后面是一组由这个领域工业界的同事向申请工作的学生提出的面试问题。这些问题可以让学生感受到面试过程中可能遇到的典型问题类型。习题的答案可以通过本书的配套网站和教师支持网站获得。

在线补充资料[Ⓔ]

补充资料可以通过 booksite.elsevier.com/9780128000564 获得。这个对所有读者开放的配套网站包括以下内容：

- 奇数编号习题的答案；
- Altera 公司专业级计算机辅助设计工具的链接；
- 链接到 Keil 的 ARM 微控制器开发套件（MDK-ARM），这是一个用于编译、汇编和模拟 ARM 处理器的 C 和汇编代码的工具；
- ARM 处理器的硬件描述语言（HDL）代码；
- 关于 Altera Quartus II 工具的提示；
- PPT 格式的电子教案；
- 简单的课程和实验素材；
- 勘误表。

教师网站包括：

- 所有习题的答案；
- 链接到 Altera 的专业级计算机辅助设计（CAD）工具；
- PDF 格式和 PPT 格式的书中插图。

在线资料提供了在课程中使用 Altera、Raspberry Pi 和 MDK-ARM 工具的指南，同时也提供了关于构建实验的详细资料。

Ⓔ 请访问华章网站 www.hzbook.com 下载在线章节。——编辑注

Ⓕ 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师请访问爱思唯尔的教材网站 <https://textbooks.elsevier.com/> 进行申请。——编辑注

如何在课程中使用软件工具

Altera Quartus II

Quartus II Web Edition 是专业级 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 模拟电路。Quartus II Web Edition 中还包含用于综合 SystemVerilog 或者 VHDL 程序的内置工具。

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

Keil 的 ARM 微控制器开发套件 (MDK-ARM)

Keil 的 MDK-ARM 是一个为 ARM 处理器开发代码的工具，可以免费下载。MDK-ARM 包括一个商业 ARM C 编译器，以及一个允许学生编写 C 和汇编程序，然后编译并模拟它们的模拟器。

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。这些实验教授学生如何使用 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 MDK-ARM 和 Raspberry Pi 开发工具完成 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 (或 DE2-115) 开发教育板上实现自己的设计。这个功能强大而且具有价格优势的开发板可以通过 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 Quartus 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 MDK-ARM (或 Raspberry Pi) 工具。教师也需要选择软件并安装在实验室的机器上。这些实验包括了如何在 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。

错误反馈

正如所有经验丰富的程序员所知，比较复杂的程序都毫无疑问地有潜在错误。这本书也不例外。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@gmail.com。第一个报告实质性错误而且在后续版本中被采用的读者可以得到 1 美元的奖励！

致谢

感谢 Nate McFadden、Joe Hayton、Punithavathy Govindaradjane 以及 Morgan Kaufmann 团队的其他成员所做的辛勤工作。

感谢 Matthew Watkins，他提供了第 7 章中关于异构多处理器的部分。非常感谢 Joshua

Vasquez 的工作，他开发了第 9 章中 Raspberry Pi 的代码。还要感谢 Josef Spjut 和 Ruye Wang，他们对资料进行了测试。

众多评审人员大大改进了这本书。他们是：Boyang Wang, 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 Teyssier, Peiyi Zhao, Zach Dodds, Nathaniel Guy, Aswin Krishna, Volnei Pedroni, Karl Wang, Ricardo Jasinski, Josef Spjut, Jörgen Lien, Sameer Sharma, John Nestor, Syed Manzoor, James Hoe, Srinivasa Vemuru, K. Joseph Hass, Jayantha Herath, Robert Mullins, Bruno Quoitin, Subramaniam Ganesan, Braden Phillips, John Oliver, Yahswant K. Malaiya, Mohammad Awedh, Zachary Kurmas, Donald Hung, 以及一位匿名审稿人。感谢 Khaled Benkrid 及其在 ARM 的同事仔细审阅了 ARM 相关材料。

非常感谢哈佛玛德学院和 UNLV 的学生，他们提供了有关本书草稿的有用反馈。特别值得一提的是 Clinton Barnes、Matt Weiner、Carl Walsh、Andrew Carter、Casey Schilling、Alice Clifton、Chris Acon 和 Stephen Brawner。

最后，但同样重要的是，感谢家人的爱和支持。

目 录

Digital Design and Computer Architecture, ARM Edition

出版者的话

赞誉

译者序

前言

第 1 章 二进制 1

1.1 课程计划 1

1.2 管理复杂性的艺术 1

1.2.1 抽象 1

1.2.2 约束 2

1.2.3 三条原则 3

1.3 数字抽象 3

1.4 数字系统 4

1.4.1 十进制数 4

1.4.2 二进制数 5

1.4.3 十六进制数 6

1.4.4 字节、半字节和字 7

1.4.5 二进制加法 8

1.4.6 有符号的二进制数 8

1.5 逻辑门 10

1.5.1 非门 11

1.5.2 缓冲 11

1.5.3 与门 11

1.5.4 或门 11

1.5.5 其他二输入逻辑门 12

1.5.6 多输入门 12

1.6 数字抽象之下 13

1.6.1 电源电压 13

1.6.2 逻辑电平 13

1.6.3 噪声容限 14

1.6.4 直流电压传输特性 14

1.6.5 静态约束 15

*1.7 CMOS 晶体管 16

1.7.1 半导体 16

1.7.2 二极管 17

1.7.3 电容 17

1.7.4 nMOS 和 pMOS 晶体管 17

1.7.5 CMOS 非门 19

1.7.6 其他 CMOS 逻辑门 19

1.7.7 传输门 21

1.7.8 类 nMOS 逻辑 21

*1.8 功耗 22

1.9 总结和展望 22

习题 23

面试问题 29

第 2 章 组合逻辑设计 30

2.1 引言 30

2.2 布尔表达式 32

2.2.1 术语 32

2.2.2 与或式 32

2.2.3 或与式 33

2.3 布尔代数 34

2.3.1 公理 34

2.3.2 单变量定理 35

2.3.3 多变量定理 36

2.3.4 定理的统一证明方法 37

2.3.5 等式化简 37

2.4 从逻辑到门 38

2.5 多级组合逻辑 40

2.5.1 减少硬件 40

2.5.2 推气泡 41

2.6 X 和 Z 43

2.6.1 非法值 X 43

2.6.2 浮空 Z 43

2.7 卡诺图 44

2.7.1 画圈的原理 45

2.7.2 卡诺图化简逻辑 45

2.7.3 无关项 48

2.7.4 小结 49

2.8 组合逻辑模块	49	*3.5.6 分辨时间的推导	95
2.8.1 多路选择器	49	3.6 并行	97
2.8.2 译码器	52	3.7 总结	99
2.9 时序	52	习题	100
2.9.1 传输延迟和最小延迟	53	面试问题	105
2.9.2 毛刺	55		
2.10 总结	57	第 4 章 硬件描述语言	107
习题	57	4.1 引言	107
面试问题	62	4.1.1 模块	107
第 3 章 时序逻辑设计	63	4.1.2 硬件描述语言的起源	108
3.1 引言	63	4.1.3 模拟和综合	109
3.2 锁存器和触发器	63	4.2 组合逻辑	110
3.2.1 SR 锁存器	64	4.2.1 位运算符	110
3.2.2 D 锁存器	65	4.2.2 注释和空格	112
3.2.3 D 触发器	66	4.2.3 缩减运算符	112
3.2.4 寄存器	67	4.2.4 条件赋值	112
3.2.5 带使能端的触发器	67	4.2.5 内部变量	114
3.2.6 带复位功能的触发器	67	4.2.6 优先级	115
*3.2.7 晶体管级的锁存器和 触发器的设计	68	4.2.7 数字	116
3.2.8 小结	68	4.2.8 Z 和 X	117
3.3 同步逻辑设计	69	4.2.9 位混合	118
3.3.1 一些有问题的电路	70	4.2.10 延迟	118
3.3.2 同步时序电路	71	4.3 结构建模	119
3.3.3 同步和异步电路	72	4.4 时序逻辑	122
3.4 有限状态机	72	4.4.1 寄存器	122
3.4.1 有限状态机设计实例	73	4.4.2 带复位功能的寄存器	123
3.4.2 状态编码	77	4.4.3 带使能端的寄存器	124
3.4.3 Moore 型状态机和 Mealy 型 状态机	79	4.4.4 多寄存器	124
3.4.4 状态机的分解	82	4.4.5 锁存器	125
3.4.5 由电路图导出状态机	83	4.5 更多组合逻辑	126
3.4.6 小结	86	4.5.1 case 语句	127
3.5 时序逻辑电路的时序	86	4.5.2 if 语句	129
3.5.1 动态约束	87	4.5.3 带有无关项的真值表	130
3.5.2 系统时序	87	4.5.4 阻塞式和非阻塞式赋值	131
*3.5.3 时钟偏移	91	4.6 有限状态机	134
3.5.4 亚稳态	92	*4.7 数据类型	137
3.5.5 同步器	93	4.7.1 SystemVerilog	137
		4.7.2 VHDL	138
		*4.8 参数化模块	140
		4.9 测试程序	142

4.10 总结	145	6.2 汇编语言	192
习题	145	6.2.1 指令	192
面试问题	152	6.2.2 操作数：寄存器、存储器和常数	193
第5章 常见数字模块	153	6.3 编程	196
5.1 引言	153	6.3.1 数据处理指令	196
5.2 算术电路	153	6.3.2 条件标志	198
5.2.1 加法	153	6.3.3 分支	200
5.2.2 减法	158	6.3.4 条件语句	201
5.2.3 比较器	159	6.3.5 循环	202
5.2.4 算术逻辑单元	160	6.3.6 存储器	204
5.2.5 移位器和循环移位器	162	6.3.7 函数调用	207
*5.2.6 乘法	162	6.4 机器语言	215
*5.2.7 除法	164	6.4.1 数据处理指令	215
5.2.8 拓展阅读	164	6.4.2 存储器指令	218
5.3 数制系统	165	6.4.3 分支指令	219
5.3.1 定点数系统	165	6.4.4 寻址模式	220
*5.3.2 浮点数系统	166	6.4.5 解释机器语言代码	220
5.4 时序电路模块	169	6.4.6 程序存储	221
5.4.1 计数器	169	*6.5 编译、汇编与加载	222
5.4.2 移位寄存器	169	6.5.1 内存映射	222
5.5 存储器阵列	171	6.5.2 编译	223
5.5.1 概述	171	6.5.3 汇编	224
5.5.2 动态随机访问存储器	173	6.5.4 链接	225
5.5.3 静态随机访问存储器	174	6.5.5 加载	226
5.5.4 面积和延迟	174	*6.6 其他主题	227
5.5.5 寄存器文件	174	6.6.1 加载文字	227
5.5.6 只读存储器	175	6.6.2 NOP	227
5.5.7 使用存储器阵列的逻辑	176	6.6.3 异常	228
5.5.8 存储器 HDL	176	6.7 ARM 体系结构的演变	230
5.6 逻辑阵列	178	6.7.1 Thumb 指令集	230
5.6.1 可编程逻辑阵列	178	6.7.2 DSP 指令	231
5.6.2 现场可编程逻辑门阵列	179	6.7.3 浮点指令	234
*5.6.3 阵列实现	182	6.7.4 节能和安全指令	235
5.7 总结	183	6.7.5 SIMD 指令	235
习题	184	6.7.6 64 位体系结构	236
面试问题	190	6.8 另一个视角：x86 体系结构	236
第6章 体系结构	191	6.8.1 x86 寄存器	237
6.1 引言	191	6.8.2 x86 操作数	238
		6.8.3 状态标志	239

6.8.4	x86 指令集	239	7.7.6	寄存器重命名	309
6.8.5	x86 指令编码	241	7.7.7	多线程	310
6.8.6	x86 的其他特性	242	7.7.8	多处理器	311
6.8.7	整体情况	242	*7.8	现实世界视角：ARM 微结构的演变	312
6.9	总结	242	7.9	总结	317
	习题	243		习题	318
	面试问题	251		面试问题	322
第 7 章	微结构	253	第 8 章	存储器系统	323
7.1	引言	253	8.1	引言	323
7.1.1	体系结构状态和指令集	253	8.2	存储器系统性能分析	326
7.1.2	设计过程	253	8.3	高速缓存	327
7.1.3	微结构	255	8.3.1	高速缓存中存放的数据	327
7.2	性能分析	255	8.3.2	高速缓存中的数据查找	327
7.3	单周期处理器	256	8.3.3	数据的替换	334
7.3.1	单周期数据通路	256	*8.3.4	高级高速缓存设计	334
7.3.2	单周期控制	261	*8.3.5	ARM 处理器中高速缓存的发展	337
7.3.3	更多指令	264	8.4	虚拟存储器	338
7.3.4	性能分析	266	8.4.1	地址转换	339
7.4	多周期处理器	267	8.4.2	页表	340
7.4.1	多周期数据通路	268	8.4.3	转换后备缓冲	341
7.4.2	多周期控制	273	8.4.4	存储器保护	342
7.4.3	性能分析	279	*8.4.5	替换策略	342
7.5	流水线处理器	281	*8.4.6	多级页表	343
7.5.1	流水线数据通路	283	8.5	总结	344
7.5.2	流水线控制	284		结语	344
7.5.3	冲突	285		习题	345
7.5.4	性能分析	293		面试问题	350
*7.6	硬件描述语言表示	293	索引		351
7.6.1	单周期处理器	294	在线章节[⊖]		
7.6.2	通用模块	298	第 9 章	I/O 系统	
7.6.3	测试程序	300	附录 A	数字系统实现	
*7.7	高级微结构	303	附录 B	ARM 指令	
7.7.1	深流水线	303	附录 C	C 语言编程	
7.7.2	微操作	304			
7.7.3	分支预测	305			
7.7.4	超标量处理器	306			
7.7.5	乱序处理器	308			

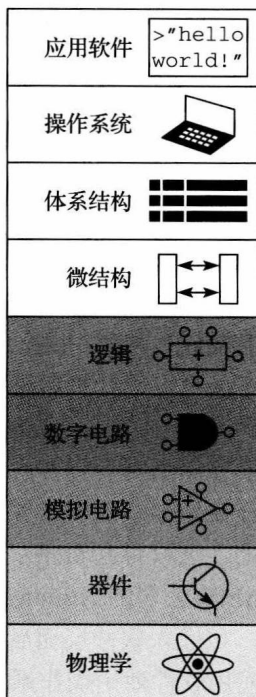
二 进 制

1.1 课程计划

在过去的 30 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大型计算机。一辆高级汽车上包含了大约 100 个微处理器。微处理器的进步使得移动电话和 Internet 成为可能，并且极大地促进了医学的进步。全球集成电路工业销售额从 1985 年的 210 亿美元发展到 2013 年的 3060 亿美元，其中微处理器占据重要部分。我们相信微处理器不仅是对技术、经济和社会有重要意义，而且潜在地激发了人类的创造力。在学习完这本书后，读者将学会如何设计和构造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浓厚的兴趣。这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，我们将学习使用以微处理器的语言——汇编语言进行程序设计。最后，我们将上述内容结合起来以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优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模块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，如何管理复杂性是贯穿全书的重要主题。



1.2 管理复杂性的艺术

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管理复杂性的方法。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。读者必须学会如何管理复杂性，从而理解如何在不断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。一个县包含了若干城市，而一个州则包含了若干县。当一个政治家竞选总统时，他对整个州的投票情况更有兴趣，而非单个县。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，因此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。电子的特征由量子力学和麦克斯韦 (Maxwell) 方程描述。我们的系统由晶体管或先前的真空管等电子器件 (device) 构造。这些器件都有明确定义的称为端子 (terminal) 的外部连接点，并建立每个端子上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 (micro-architecture) 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 computers, PC) 的 Intel 公司的 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)，从而程序员可以使用这些指令和寄存器。微结构将逻辑组件组合在一起以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，以取得在价格、性能和功耗等方面的不同折中。例如，Intel 公司的 Core i7、80486 和 AMD 公司的 Athlon 等都是 x86 体系结构的不同微结构实现。

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以解决用户的问题。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或计算机中的存储器组织问题。

这本书将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，能选择正确的层次以解决问题，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要弱于范围更广的模拟

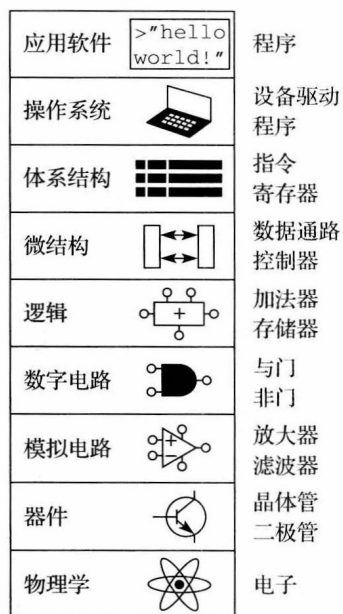


图 1-1 电子计算机系统的抽象层次

4

电路。相对而言，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。

5

1.2.3 三条原则

除了抽象和约束外，设计者还使用另外三条原则来管理系统的复杂性：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，直到这些模块都很容易理解。
- 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，通用的模块可以重新使用多次，以减少设计不同模块的数量。

我们通过制作来复枪的例子来解释这三条原则。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，我们可以将它划分为图 1-2 所示的几个组件：枪机、枪托和枪管。

枪管是一个长金属管子，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，它将各种部件连接起来并且为用户提供牢固的握枪位置。更进一步，枪机包含扳机、击锤、燧石、扣簧和药锅。对每种组件都可以展开更详细的层次化描述。

模块化使得每个组件都有明确的功能和接口。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，只要枪托和枪管长度正确并有适当的安装机制，那么来自不同制造商的枪托就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对其他部件产生影响：对枪托的设计不能影响到枪管的功能。

规整化表明可互换部件是一个好方法。利用规整化原则，损坏的枪管可以用相同的部件取代。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、模块化和规整化三条原则在本书中很重要，它们将贯穿本书。

1.3 数字抽象

大部分物理变量是连续的，例如电线上的电压值、振动的频率、物体的位置等都是连续的值。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。

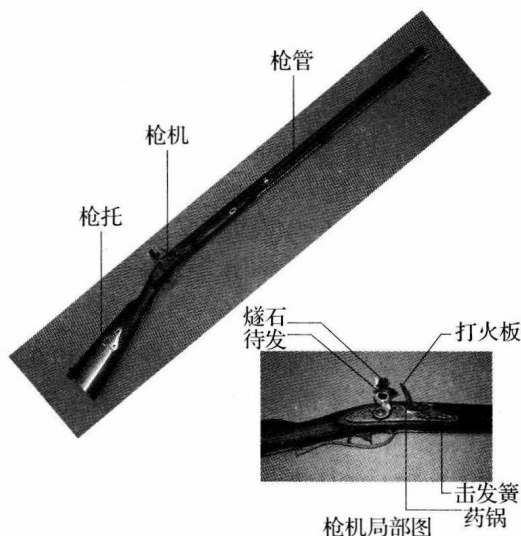


图 1-2 燧石来复枪及其枪机的特写照片（图片来源：意大利 Euroarms, www.euroarms.net © 2006）

6