

HZ BOOKS
华章教育

计 算 机 科 学 丛 书

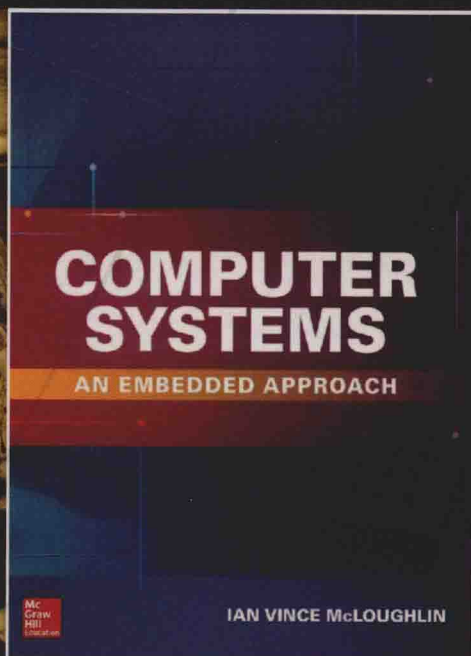
Mc
Graw
Hill

计算机系统 嵌入式方法

[英] 伊恩·文斯·麦克洛克林 (Ian Vince McLoughlin) 著

刘雯 译

探索全面推动从微型嵌入式设备到全球物联网的现代计算机系统技术



Computer Systems
An Embedded Approach



机械工业出版社
China Machine Press

计

算

人

书

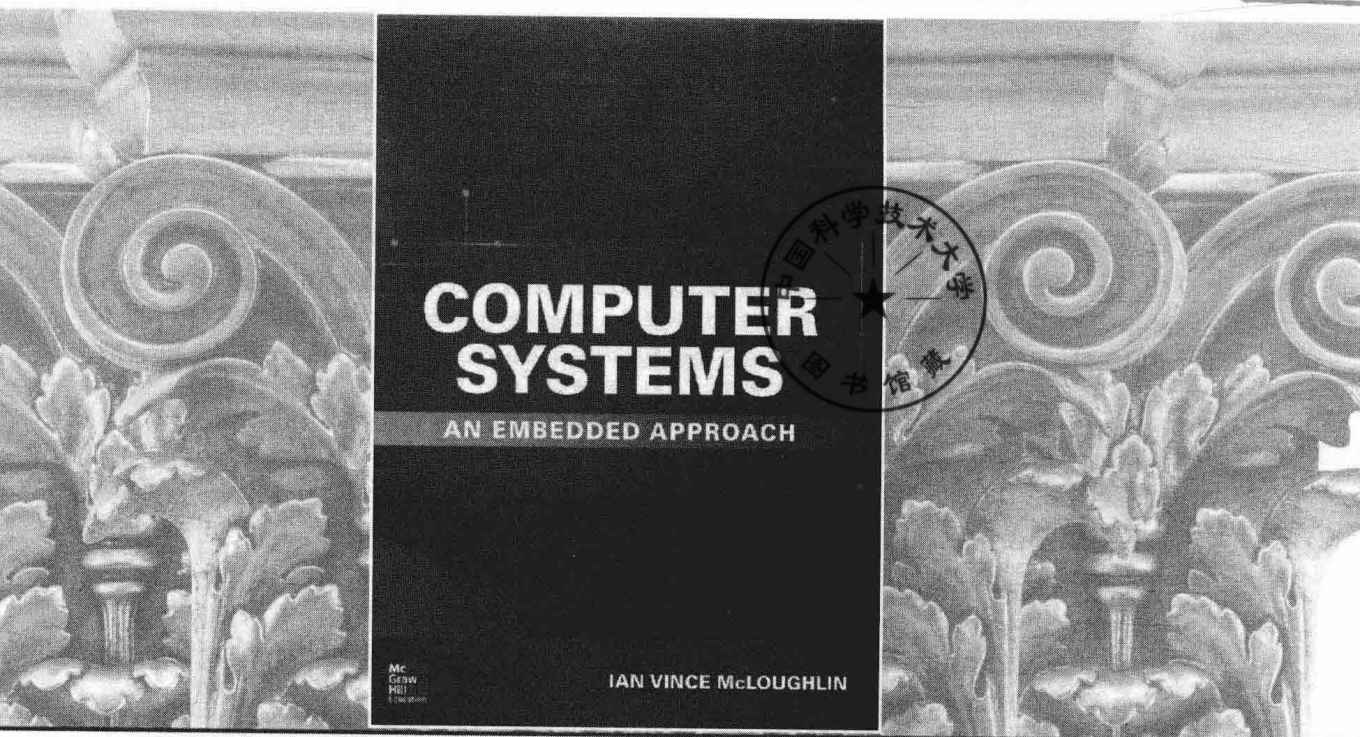
计算机系统 嵌入式方法

[英] 伊恩·文斯·麦克洛克林 (Ian Vince McLoughlin) 著

刘雯 译

Computer Systems

An Embedded Approach



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

计算机系统：嵌入式方法 / (英) 伊恩·文斯·麦克洛克林 (Ian Vince McLoughlin) 著；刘雯译. —北京：机械工业出版社，2020.6

(计算机科学丛书)

书名原文：Computer Systems: An Embedded Approach

ISBN 978-7-111-65722-4

I. 计… II. ①伊… ②刘… III. 计算机系统 IV. TP303

中国版本图书馆 CIP 数据核字 (2020) 第 090219 号

本书版权登记号：图字 01-2019-0935

Ian Vince McLoughlin

Computer Systems: An Embedded Approach

978-1-260-11760-8

Copyright © 2018 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and China Machine Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2020 by McGraw-Hill Education and China Machine Press.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和机械工业出版社合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。

版权 © 2018 由麦格劳-希尔(亚洲)教育出版公司与机械工业出版社所有。

本书封面贴有 McGraw-Hill Education 公司防伪标签，无标签者不得销售。

这是一本从嵌入式角度探索计算机硬件和软件原理的综合教科书。本书逐步揭示了如何在现实世界中运用这些计算原理构建计算机系统，包括从小型嵌入式设备到仓库大小的计算机集群的各种计算机系统，并揭示了这些概念是如何通过互联网在全球范围内相互关联的。本书首先完整地讲述了主要的硬件组件，包括处理器、内存、存储设备与性能加速器，接着充分探讨了操作系统、连接以及网络。通过本书，你将了解计算机硬件和软件是如何协同工作来支持无处不在的计算、物联网、移动计算技术以及大小不一的应用的。

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：赵亮宇

责任校对：李秋荣

印刷：北京瑞德印刷有限公司

版次：2020 年 6 月第 1 版第 1 次印刷

开本：185mm × 260mm 1/16

印张：25.25

书号：ISBN 978-7-111-65722-4

定价：139.00 元

客服电话：(010) 88361066 88379833 68326294

投稿热线：(010) 88379604

华章网站：www.hzbook.com

读者信箱：hzsj@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



华章科技图书出版中心

人工智能与智能硬件离我们的生活越来越近了，我们在日常生活的方方面面都能感受到这些技术带来的便利，然而这些对我们的生活产生潜移默化影响的科技还是需要以计算机为载体。目前国内很多学生受就业热潮的影响选择计算机专业，但大多偏向于应用算法的研究，对计算机系统的基础理论重视不够，这在一定程度上阻碍了计算机事业的发展。本书是一本对计算机系统解读、梳理得比较透彻的书籍。伊恩·文斯·麦克洛克林教授是英国查塔姆肯特大学梅德韦校区现任计算机学院院长、计算机教授，还是 IET 的成员、IEEE 高级会员、英国的特许工程师和欧洲（欧盟）工程师，在计算机系统，尤其是嵌入式技术应用方面颇有建树，同时还有丰富的工程经验。本书内容由浅入深，既有对计算机系统软硬件基础详细的入门介绍，又有结合实际应用，对嵌入式系统技术存在的瓶颈的深入探讨，适合作为计算机及相关专业本科生的教材或教学参考书，同时适合其他热爱计算机科学技术，希望进一步提升自己的计算机基础和能力的读者阅读参考，甚至对于工作在嵌入式系统产业的工程师来说，本书也是一本不错的参考书。

本书将计算机技术的发展背景与基础知识作为切入点，以嵌入式系统的方向为特色，涉及计算机系统内部包含的硬件、软件编程与操作系统、连接性与网络系统。硬件部分结合嵌入式系统与实际应用对 CPU 的功能、内部组成单元以及计算机接口进行详细介绍。软件部分包括程序设计、工作原理以及操作系统。连接性与网络系统部分介绍计算机网络发展至今的拓扑与连接方法（网络协议等）。最终以对计算机系统的未来展望作为结尾。纵观全书，作者用通俗易懂的语言将嵌入式开发设计流程与核心技术进行了详细介绍。与其他高等教育专业教科书不同，作者的叙述风格诙谐幽默，并且结合了实际应用案例，十分便于读者理解，字里行间可以感受到作者极高的写作热情、深厚的学术背景与丰富的嵌入式系统开发经验。

当朱捷编辑将这本教材推荐给我，并邀请我作为中文版的译者时，我还是心怀敬畏的，日常科研教学工作任务繁重，自己是否能再接下这本书的翻译工作？通过对原著的阅读以及与编辑的沟通，我认为本书是基于嵌入式系统架构讨论计算机体系的一本好书，值得推荐给我们的学生和嵌入式系统的从业工程师。在翻译过程中力争既要忠实于原著，又要尊重中文的表达方式，尤其要将英文中的幽默语气用中文表达方式传达给读者，实践起来也颇为不易。在这里要感谢我的同事徐国鑫老师，他对“嵌入式原理”这门课程教学的一些需求及对原著的肯定促使我下决心开始翻译本书。我的学生程倩倩、莫耀凯、隋钰童、王博斐、贾铭杰分担了许多协助翻译、查找资料及校对工作。他们的付出与努力是整个翻译工作能够顺利完成的强有力的保障！同时也感谢朱捷编辑在整个过程中的耐心指导和悉心审阅。

本书的翻译工作历时近 7 个月，每章译稿都经过至少两个人的多遍校阅。作为译者，我们努力保证译本的准确、易读，但由于时间紧、任务急，译作之中难免存在疏漏之处，恳请各位同行及广大读者批评指正。希望我们努力的结果能够使读者满意，同时也希望本书中文版的出版能为嵌入式系统技术在中国的发展贡献力量。

智能手机、便携式游戏设备等都是广义上的计算机，它们在人们的日常生活中正变得越来越重要。本书致力于从多个层面理解这些系统，让读者明白是什么使其成为计算机，并强调它们背后的内核——嵌入式系统，以及嵌入式系统真正令人着迷的技术。审视嵌入式系统的内部，对于一个有技术头脑的人来说，就像打开了一个充满知识的圣诞礼物。

书店（特别是大学城里的书店）里关于这个主题的教科书似乎随处可见，比如计算机体系结构、计算机系统设计、网络、操作系统，甚至是嵌入式系统。许多著名的技术作者也都尝试过在这个领域写作，但计算机是一个不断发展的技术领域，很难用会很快过时的静态教科书充分地描述。特别是在过去十年，嵌入式计算系统的兴起让一些思维保守的作者感到惊讶：在20世纪五六十年代的一些教科书中，人们坚持将计算机视为房间大小的机器，20世纪八九十年代的教科书将计算机视为台式和服务器计算机，只有少数作者真正认为未来使用的绝大多数计算机都是嵌入在日常物品中的，认为计算机在未来是嵌入式的、互通的和无处不在的。在未来，台式机甚至笔记本电脑将会像50年前的打孔机一样不合时宜。

本书中的讨论尽可能直接面向嵌入式，并使用来自嵌入式世界的例子，这些例子适用于计算机体系结构、操作系统和连接这三个子领域。有些主题与嵌入式处理器有更紧密的联系，其他的可能是更传统的主题。这里将尽可能给出嵌入式领域的一些例子，以及可以描述章节主题与读者的联系的相关资料。

全书结构

本书由12章构成，除了第1、2章的介绍与第12章的总结部分之外，其他章节可以粗略分为三个部分，依次解决以下三个问题：

- 现代计算机系统（嵌入式或其他系统）所用的内部硬件，其工作原理以及协同方法。
- 在使用计算机编程实现功能时需要怎么做？软件如何被写入、加载并执行？如何系统化并呈现给用户？现代计算机内部的系统如何管理？
- 计算机之间如何通过连接来交换信息并为用户提供分布式服务？

总的来说，在“引言”和“基础知识”章节之后，第3~7章会讨论硬件，相关的软件编程和操作系统知识将在第8章和第9章介绍，连接性和网络系统的内容在第10章和第11章介绍。本书各部分之间紧密联系，但是读者不必按顺序学习本书——这三部分都包含对其他部分的介绍，非常通俗易懂。

适用读者

本书适用于计算机、计算机科学、计算机工程、计算机系统工程、电子与计算机工程、电子与电气工程等专业的本科生。对于工作在其他技术领域的人来说，如果对学习计算机基础有兴趣，本书也是不错的选择。引言与基础知识部分适合本科生前两年学习，本书还有足够的深度与拓展部分，适合学习计算机架构或计算机系统设计的大四学生阅读。

对于从事嵌入式系统相关工作的程序员与工程师来说，本书是十分有用的参考。本书对各个行业都十分适用的ARM处理器的重点介绍一定会受到业内人士的欢迎。

本书设计

本书的内容是自下而上编写的，除了本书的硬件和计算机体系结构章节的某些部分以我以前的作品《计算机体系结构：嵌入式方法》（*Computer Architecture: An Embedded Approach*）为基础以外，其他部分并没有借鉴现有的任何教科书。我采用了一种新的方法，并在不受传统结构约束的情况下规划本书，避开了计算机进化过程中出现的“死胡同”以及一些不相关的问题。这让读者的思路更加清晰，可以将注意力集中在嵌入式系统上（本书没有完全略去大型机器，因为它们包含了多年来已普及的更重要且更好的技术演变的优秀例子）。

在创作本书的过程中，我的目的是通过编写简单易懂的文字来培养读者的兴趣，并努力保持本书与影响日常生活的流行前沿技术的相关性。然而，在任何研究领域都存在一些棘手的概念，遇到这些概念时，我会写出明确的解释性文字，同时在许多地方提供了充分而直观的说明来帮助读者理解。此外，本书还提供了许多解释框，这些解释框包含了额外的示例、有趣的信息和其他解释之类的材料，目的是扩充正文内容并帮助读者吸收知识。

本书使用 SI（国际单位制），包括新的计算机存储度量单位 kibibyte 和 mebibyte（附录 A 中对此进行了解释）。本书的主要章节附有思考题。[⊖]

本书的编写

书中的示例代码和大多数嵌入式系统描述源自作者的硬件和软件设计经验。非常感谢 GNU 项目中出色的 GCC ARM 编译器，以及 Busybox（嵌入式系统编码的“瑞士军刀”）和 ARM / Linux 项目。

本书中的一些图像由 Wikimedia Commons 等在线资源提供，均在图题中有所标注。这些图像均有知识共享（CC）归属（BY）许可或共享（SA）许可。特别要感谢 Wikimedia Commons 提供了优秀资源以及 Creative Commons[⊖]提供了保护许可证。

开始之前

我们应该铭记数十年来辛勤工作的计算机工程师们，他们为现代社会的移动电话、智能机、计算机和嵌入式技术的蓬勃发展注入了重要的力量。我们赞扬过去那些伟大的贡献，并希望读者能够畅游在嵌入式计算机系统知识的海洋里，努力建设更美好的科技未来。

Ian Vince McLoughlin

⊖ 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，需要的教师可向麦格劳-希尔教育出版公司北京代表处申请，电话 010-57997618/7600，传真 010-59575582，电子邮件 instructorchina@mhedu.com。——编辑注

⊖ 可以在 <https://creativecommons.org/licenses> 上查看本书中所使用的 CC 图像的完整许可文件。

在此特别感谢我耐心的妻子 Kwai Yoke、可爱的孩子 Wesley 和 Vanessa，他们给了我充足的时间撰写此书。而纽约的 McGraw-Hill 教育的所有编辑和制作人员，则在我需要的时候给了我很多鼓励，他们的专业精神促成了本书的出版。同时，也感谢 Gerald Bok 和其他在新加坡 McGraw-Hill 亚洲团队工作的人，特别是 Gerald 在 2006 ~ 2013 年对我写作生涯的支持。

我还要在此感谢很多朋友的支持和鼓励，但我更想把这份成果献给我的母亲，不仅仅是在写这本书时，更是在我的整个生命历程中，是她始终鼓励着我，不离不弃，风雨相随。正是她对我的殷切期望，使我最终走上科学研究的道路，而她也始终以最饱满的热情来关注我对于写作的尝试。

关于作译者

Computer Systems: An Embedded Approach

关于作者

伊恩·文斯·麦克洛克林是英国查塔姆的肯特大学梅德韦校区现任计算机学院院长、计算机教授。在其 30 多年的职业生涯中，他一直致力于在三大洲的工业、政府和学术领域工作并强调研究和创新。他是一名计算机工程师，曾经参与设计对流层和太空中的计算系统，以及在水下的电信网络，在民用方面，曾将嵌入式设备应用到喉咙手术上来帮助病人说话。他还是 IET 成员、IEEE 高级会员、英国注册工程师和全欧工程师。

关于译者

刘雯，任教于北京邮电大学电子工程学院，博士生导师，主要研究方向为室内外高精度定位技术及位置服务。主持国家重点研发计划课题 1 项、国家自然科学基金项目 1 项，主持或参与完成国家 863 课题多项；获得国家科技发明二等奖 1 项、国家科技进步二等奖 1 项，省部级一、二等奖 8 项；出版多本嵌入式方向教材。

出版者的话
译者序
前言
致谢
关于作译者

第1章 引言 1

- 1.1 计算机的进化 1
- 1.2 进化过程 1
- 1.3 计算机发展阶段划分 4
 - 1.3.1 第一代计算机 4
 - 1.3.2 第二代计算机 5
 - 1.3.3 第三代计算机 5
 - 1.3.4 第四代计算机 6
 - 1.3.5 第五代计算机 7
- 1.4 云、普适、网格和超并行计算机 7
- 1.5 未来 8
- 1.6 小结 10

第2章 基础知识 11

- 2.1 计算机组成 11
 - 2.1.1 Flynn 分类法 11
 - 2.1.2 连接方式 12
 - 2.1.3 计算机结构层次视图 13
- 2.2 计算机基本原理 13
- 2.3 数字格式 16
 - 2.3.1 无符号二进制 16
 - 2.3.2 原码 17
 - 2.3.3 反码 17
 - 2.3.4 补码 17
 - 2.3.5 移码 18
 - 2.3.6 BCD 码 19
 - 2.3.7 定点数表示法 19
 - 2.3.8 符号扩展 20
- 2.4 算术运算 20
 - 2.4.1 加法 20
 - 2.4.2 并行进位传递加法器 20

- 2.4.3 超前进位 22
- 2.4.4 减法 22
- 2.5 乘法 24
 - 2.5.1 加法迭代法 24
 - 2.5.2 部分积方法 25
 - 2.5.3 移位加方法 27
 - 2.5.4 Booth 和 Robertson 方法 27
- 2.6 除法 29
- 2.7 定点数格式的运算 30
 - 2.7.1 定点数的运算 31
 - 2.7.2 定点数的乘除 32
- 2.8 浮点数 32
 - 2.8.1 广义浮点数 33
 - 2.8.2 IEEE754 浮点标准 33
 - 2.8.3 IEEE754 标准模式 34
 - 2.8.4 IEEE754 数的范围 36
- 2.9 浮点数处理 38
 - 2.9.1 IEEE754 数的加减运算 39
 - 2.9.2 IEEE754 数的乘除运算 41
 - 2.9.3 IEEE754 中间格式 41
 - 2.9.4 舍入 42
- 2.10 小结 42
- 思考题 43

第3章 CPU 基础 45

- 3.1 什么是计算机 45
- 3.2 让计算机为你服务 45
 - 3.2.1 程序存储 46
 - 3.2.2 存储架构 46
 - 3.2.3 程序传输 47
 - 3.2.4 控制单元 48
 - 3.2.5 微指令 52
 - 3.2.6 RISC 和 CISC 的对比 53
 - 3.2.7 处理器实例——ARM 55
 - 3.2.8 关于 ARM 的更多内容 56
- 3.3 指令处理 56
 - 3.3.1 指令集 57

3.3.2	取指和译码	59	4.4.6	cache 性能	104
3.3.3	压缩指令集	63	4.4.7	cache 一致性	105
3.3.4	寻址模式	65	4.5	协处理器	107
3.3.5	堆栈机和逆波兰表示法	67	4.6	浮点运算单元	107
3.4	数据处理	68	4.7	SIMD 流指令扩展和多媒体 扩展	109
3.4.1	数据的格式和表达	68	4.7.1	MMX	109
3.4.2	数据流	71	4.7.2	MMX 实现	110
3.4.3	数据存储	72	4.7.3	MMX 的使用	111
3.4.4	内部数据	72	4.7.4	SIMD 流指令扩展	111
3.4.5	数据处理	73	4.7.5	使用 SSE 和 MMX	111
3.5	自顶向下方法	75	4.8	嵌入式系统中的协处理	112
3.5.1	计算机的能力	75	4.9	小结	112
3.5.2	性能衡量和统计	76		思考题	113
3.5.3	性能评估	78			
3.6	小结	79			
	思考题	80			
第 4 章	处理器内部组成	82	第 5 章	提高 CPU 性能	116
4.1	内部总线结构	82	5.1	加速	116
4.1.1	程序员的角度	82	5.2	流水线	117
4.1.2	分解互联排列	83	5.2.1	多功能流水线	118
4.1.3	ADSP21xx 总线排列	84	5.2.2	动态流水线	119
4.1.4	数据与程序同时访存	84	5.2.3	改变流水线模式	119
4.1.5	双总线体系结构	86	5.2.4	数据相关冒险	121
4.1.6	单总线体系结构	87	5.2.5	条件冒险	122
4.2	算术逻辑单元	88	5.2.6	条件分支	123
4.2.1	ALU 功能	88	5.2.7	编译时流水线补偿	125
4.2.2	ALU 设计	89	5.2.8	相对地址分支	126
4.3	内存管理单元	91	5.2.9	流水线的指令集补偿	127
4.3.1	对虚拟存储的需求	91	5.2.10	运行时流水线补偿	128
4.3.2	MMU 操作	91	5.3	复杂指令集和精简指令集	130
4.3.3	退回算法	93	5.4	超标量体系结构	130
4.3.4	内部存储碎片和片段	93	5.4.1	简单超标量	130
4.3.5	外部碎片	94	5.4.2	多发送超标量	132
4.3.6	改进的 MMU	95	5.4.3	超标量的性能	133
4.3.7	内存保护	96	5.5	每周期的指令数	133
4.4	cache	97	5.5.1	不同体系结构的 IPC	133
4.4.1	直接相联 cache	98	5.5.2	IPC 度量	134
4.4.2	组相联 cache	99	5.6	硬件加速器	135
4.4.3	全相联 cache	100	5.6.1	零开销循环	135
4.4.4	局部性原则	100	5.6.2	地址处理硬件	137
4.4.5	cache 替换算法	102	5.6.3	影子寄存器	140
			5.7	分支预测	140
			5.7.1	分支预测的必要性	141

5.7.2	单T位预测器	142	6.4.3	实时性定义	184
5.7.3	双位预测器	143	6.4.4	时间范围参数	185
5.7.4	计数器和移位器预测器	145	6.4.5	硬件体系结构对实时操作 系统的支持	186
5.7.5	局部分支预测器	145	6.5	中断和中断处理	187
5.7.6	全局分支预测器	148	6.5.1	中断的重要性	187
5.7.7	G选择预测器	149	6.5.2	中断过程	187
5.7.8	G共享预测器	150	6.5.3	高级中断处理	191
5.7.9	混合预测器	151	6.5.4	共享中断	191
5.7.10	分支目标缓冲	152	6.5.5	可重入代码	192
5.7.11	基本代码段	153	6.5.6	软件中断	192
5.7.12	分支预测总结	154	6.6	嵌入式无线连接	192
5.8	并行机器	155	6.6.1	无线技术	192
5.8.1	SISD向MIMD的演变	157	6.6.2	无线接口	194
5.8.2	为提高性能而采用并行	159	6.6.3	无线相关问题	194
5.8.3	其他并行处理	160	6.7	小结	194
5.9	Tomasulo算法	163	思考题		195
5.9.1	Tomasulo算法的原理	163	第7章	实用嵌入式CPU	198
5.9.2	Tomasulo系统的例子	164	7.1	概述	198
5.9.3	嵌入式系统中的Tomasulo 算法	167	7.2	微处理器不只是核	198
5.10	超长指令架构集	168	7.3	功能需求	200
5.10.1	什么是VLIW	168	7.4	时钟	203
5.10.2	VLIW的优势	169	7.5	时钟与功耗	205
5.10.3	VLIW的瓶颈	170	7.5.1	传输延迟	206
5.10.4	与超标量处理器的比较	170	7.5.2	电流相关问题	206
5.11	小结	171	7.5.3	时钟问题解决方法	207
思考题		171	7.5.4	低电压设计	207
第6章	外部总线	174	7.6	存储	208
6.1	总线接口	174	7.6.1	早期的计算机存储	208
6.1.1	总线控制信号	175	7.6.2	只读存储器	209
6.1.2	直接存储器存取	175	7.6.3	随机存取存储器	213
6.2	并行总线规范	176	7.7	分页与重叠	220
6.3	标准接口	177	7.8	嵌入式系统中的存储	221
6.3.1	系统控制接口	177	7.8.1	非易失存储器	222
6.3.2	系统数据总线	178	7.8.2	其他存储器	224
6.3.3	输入/输出总线	182	7.9	测试和验证	225
6.3.4	外设器件总线	182	7.9.1	集成电路设计和制造 问题	225
6.3.5	与网络设备的接口	183	7.9.2	BIST	226
6.4	实时性问题	183	7.9.3	JTAG	228
6.4.1	外部激励	184	7.10	错误检测和纠正	230
6.4.2	中断	184			

7.11 看门狗定时器和复位监测	233	9.3.5 存储和归档	274
7.12 逆向工程	235	9.3.6 保护和错误处理	274
7.12.1 逆向工程过程	236	9.4 操作系统的结构	275
7.12.2 详细的物理布局	239	9.4.1 分层操作系统	276
7.13 防止逆向工程	242	9.4.2 客户端-服务器操作系统	277
7.13.1 存储程序的被动模糊	243	9.5 启动	277
7.13.2 可编程逻辑家族	244	9.5.1 从并行闪存启动	278
7.13.3 主动 RE 防范	244	9.5.2 从 HDD/SSD 启动	279
7.13.4 主动 RE 防范分类	245	9.5.3 启动之后	280
7.14 软核处理器	246	9.6 进程	281
7.14.1 微处理器不仅仅是核心	246	9.7 调度	283
7.14.2 软核处理器的优点	246	9.8 存储与文件系统	286
7.15 硬件软件协同设计	248	9.8.1 二级存储	286
7.16 商业处理器内核	251	9.8.2 文件系统的作用	289
7.17 小结	252	9.8.3 什么是文件系统	291
思考题	252	9.8.4 备份	297
第 8 章 编程	254	9.9 小结	298
8.1 运行一个程序	254	思考题	298
8.1.1 执行的含义	255	第 10 章 连接性	300
8.1.2 注意事项	257	10.1 连接的原因与方法	300
8.2 编写程序	258	10.1.1 一对一通信	300
8.2.1 编译型语言	258	10.1.2 一对多通信	301
8.2.2 解释型语言	261	10.1.3 包交换	302
8.3 UNIX 编程模型	263	10.1.4 简单通信拓扑	303
8.3.1 shell	263	10.2 系统要求	304
8.3.2 重定向和数据流	264	10.2.1 分组化	304
8.3.3 实用软件	266	10.2.2 编码与解码	305
8.4 小结	266	10.2.3 传输	305
思考题	266	10.2.4 接收	305
第 9 章 操作系统	268	10.2.5 错误控制	306
9.1 操作系统的含义	268	10.2.6 连接管理	309
9.2 为什么需要操作系统	268	10.3 可扩展性、效率与重复利用	310
9.2.1 操作系统的特征	269	10.4 OSI 分层	311
9.2.2 操作系统的类型	270	10.5 拓扑与架构	311
9.3 操作系统的作用	271	10.5.1 分层网络	312
9.3.1 资源管理	271	10.5.2 主从架构	312
9.3.2 虚拟机	271	10.5.3 对等架构	313
9.3.3 CPU 时间	272	10.5.4 点对点连接	313
9.3.4 内存管理	273	10.5.5 移动性与切换	313
		10.6 小结	314
		思考题	314

第 11 章 网络系统	316	11.9 小结	337
11.1 因特网	316	思考题	337
11.1.1 因特网的历史	317	第 12 章 未来	338
11.1.2 因特网治理	317	12.1 单比特结构	338
11.2 TCP/IP 和 IP 层模型	318	12.1.1 比特 - 串行加法	339
11.3 以太网概述	321	12.1.2 比特 - 串行减法	340
11.3.1 以太网数据格式	322	12.1.3 比特 - 串行逻辑和处理 ...	340
11.3.2 以太网封装	323	12.2 多并行机器	341
11.3.3 以太网载波侦听	324	12.2.1 小型 CPU 集群	341
11.4 网络层	325	12.2.2 并行和集群处理注意 事项	345
11.4.1 IP 地址	325	12.2.3 互连策略	345
11.4.2 网络数据包格式	326	12.3 异步处理器	347
11.4.3 路由	327	12.3.1 数据流控制	348
11.4.4 单播与多播	327	12.3.2 避免管道冒险	349
11.4.5 任播	328	12.4 替代数字格式系统	349
11.4.6 命名	328	12.4.1 多值逻辑	350
11.4.7 域名服务器	329	12.4.2 有符号数字表示	350
11.5 传输层	331	12.5 光学计算	353
11.5.1 端口号	331	12.5.1 电 - 光全加器	353
11.5.2 UDP	332	12.5.2 电 - 光底板	354
11.5.3 TCP	332	12.6 是科幻小说还是未来的现实 ...	355
11.5.4 UDP 与 TCP 对比	333	12.6.1 分布式计算	356
11.6 其他信息	333	12.6.2 湿件	356
11.6.1 地址解析协议	333	12.7 小结	357
11.6.2 控制信息	334	附录 A 标准内存大小表示方法	358
11.7 无线连通性	334	附录 B 标准逻辑门	360
11.7.1 WiFi	334	索引	361
11.7.2 WiMax	335		
11.7.3 蓝牙	335		
11.7.4 ZigBee	336		
11.7.5 近场通信	336		
11.8 网络量表	336		

1.1 计算机的进化

计算机进化经历了一个漫长的历程：从1834年 Charles Babbage 的差分机（见图1-1，它与后来的数学计算处理机具有类似设计）到今天的超级计算机，是一段为提高处理能力、复杂性和微型化而不懈努力的佳话。

令人吃惊的是，Babbage 当年的很多技术（以及20世纪40年代的早期电子计算机）非常具有前瞻性，在当今计算机系统中依然随处可见。这显示了早期的开拓者惊人的远见，同时也证明了一些基本的运作和结构几乎在所有类型的计算机中都是通用的，不分年代。今天我们有机会在事后通过计算发展的历史进行反思，从而可以发现许多生命周期不长的进化分支，这些分支最初看上去很有前途，但很快失去了生命力。它们当中有些可能会在后来的一些专门化机器中重新出现，但更多的仅仅成了一段历史。

未来的计算机很有可能会建立在当前所用的技术之上。当前技术的快照（正如所有计算机教科书所必须做的）要如实反映这个事实，而不能脱离历史地介绍技术。

本书将按照技术进化过程展开。前面几章重点描述计算机基础。掌握这些基础能使学生们在纸面上构造出可工作但效率低的计算机。之后的章节将描述当今体系结构在提高性能方面的先进技术。我们将这些先进技术与计算机基础分开讲述，原因在于它们当中有些可能是进化过程中的“死胡同”，仅在当今推动 Moore 定律[⊖]快速向前。

在计算机体系结构方面最终会出现革命性的变化——它将打破进化趋势，使得许多曾经有助于提高性能的技术被遗忘。本书中不能确定这些技术是什么，但可以做一些非正式的猜测。在本书的最后一章，我们将讨论在未来几十年可能带来革命性变化的先进技术。虽然我们不能确定未来会出现哪些颠覆性的技术，但是可以明确过去和现在的计算机在性能方面的提升。最重要的是，可以理解驱动架构师、计算机系统设计师、软件开发人员、网络工程师在从事的工作。

1.2 进化过程

计算机的发展遵循着不断提升的进化路线。其中颠覆性的突破虽然很罕见，但一直备受欢

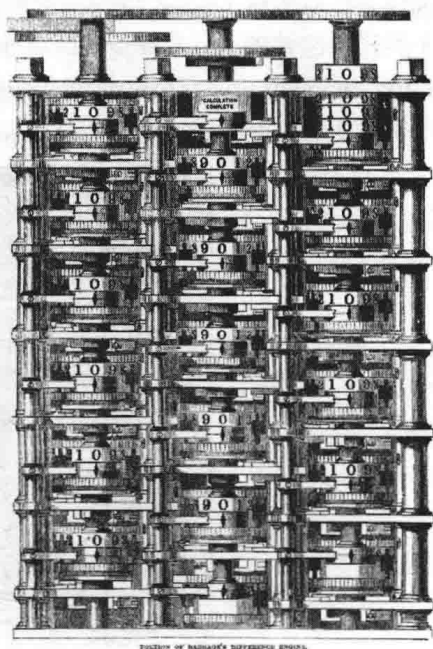


图1-1 Babbage 的差分机的一部分，来源于 Harper 的月刊杂志第30卷，第175期，p34，1864年。在伦敦科学博物馆可以见到其原始文档和一台可工作的重建机器

⊖ 英特尔的工程师戈登·摩尔（Gordon Moore）在1965年指出，晶体管密度（以及因此导致的计算机复杂度）正以指数级的速度增长。他预测，这种增速将在未来持续下去，而这一想法随后被称为摩尔定律。

迎。近些年来，计算机领域都在小幅提升。

当然，像计算机这样复杂的设备需要由有才华的工程师来设计。我们能够记得一些工程师的名字，尤其是那些做出重大改进的人（其中有些人还在世）。另外，那些开创性机器的设计和历史往往付出了巨大代价，应当被很好地记录下来。

因此，人们往往希望计算机的发展历史是清晰定义的，对半个世纪以来出现的那些开创性计算机的认识没有困惑和争议。但遗憾的是实际情况并不是这样：存在着非常不同的意见，关于准确的时间、贡献、“第一”等几乎没有一致意见。任意比较两本关于计算体系结构或计算机历史的书就会发现这一点。出于本书目的，我们将从巨人 Colossus 开始展现计算机历史。

Colossus（如图 1-2 所示）由工程师 Tommy Flower 于 1943 年制造，由 Alan Turing 及其在 Bletchley Park 的同事完成程序设计，目前普遍认为这是世界上第一台可编程电子计算机。这台计算机作为针对德国 Enigma 编码的密码破译工程（最终成功）的一部分，于第二次世界大战中构建于英国。遗憾的是，Colossus 因属于英国官方保密法之列而被隐藏了整整 50 年。战后，首相 Winston Churchill（以典型的保密文件的形式）命令将这台机器拆分成不大于人手的碎片，同时命令销毁所有与 Colossus 相关的文章。规划和图纸由设计者焚烧，解密操作员宣誓保守秘密。

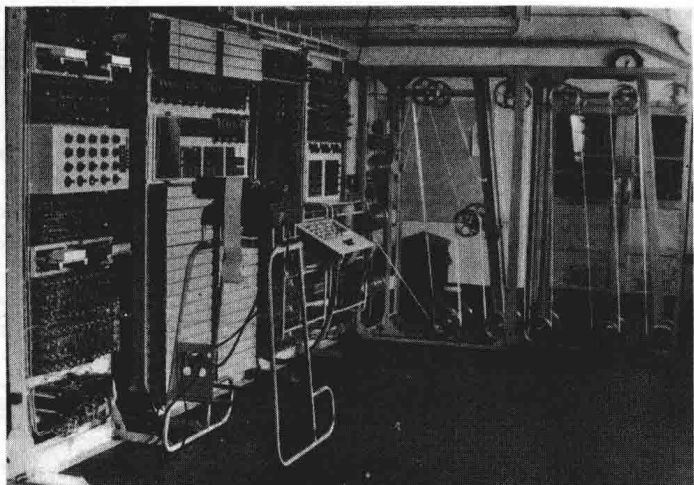


图 1-2 照片 1945 年拍摄于 Bletchley Park 的 H 区，是当时使用的十台 Colossus 计算机之一。国家计算机博物馆的洛仑兹密码机现在存放在这里（图片由英国档案馆提供）

这台机器被隐藏得很成功。尽管偶尔有一些关于这台机器的传闻，但公众最终是在 2000 年所剩无几的几份相关文件被解密时才得知 Colossus 的存在。因此，在很多计算机历史的相关描述中都没有提到 Colossus，整整一代计算机设计师从来没有听说过它。

然而，另有一些著名的与 Colossus 同期的机器在之后几年开始使用。其中最著名的 ENIAC (Electronic Numerical Integrator And Computer) 在美国制造。当 Colossus 还在被完全隐藏时，1944 年开始投入使用的 ENIAC 显然成为世界上第一台数字计算设备。许多不知道 Colossus 的教科书作者将 ENIAC 作为第一台现代计算机。事实上，Colossus 不仅投入使用的更早，而且与现代计算机相似，是一台二进制计算机，而 ENIAC 是一台十进制计算机。但 Colossus 和 ENIAC 一样都很难编程，需要通过调整开关和变换连线位置实现重新编程。

令人惊叹的是，出现在一个世纪之前的 Charles Babbage 的差分机是数字的而不是模拟的，且完全可编程，比起那些被称为第一的电子计算机在某些方面更为先进。Babbage 还设计了一台打印机外设，可以输出数字计算结果。Babbage 的机器还具有完整的程序设计语言，能够处理循环和条件分支，这使得 Babbage 的朋友 Lovelace 伯爵夫人 Ada Byron（著名诗人拜伦的女儿）在这台计算机上工作并写出了世界上第一个计算机程序。这可能是历史上诗歌创作和编程

第一次也是最后一次走到一起。

在差分机和 Colossus 之间，计算领域并不完全是荒漠：德国人 Konrad Zuse 在 1940 ~ 1941 年间构建了一台电力计算机，归类为电力而不是电子计算机的原因是它基于继电器建造而成。另一个电子计算机的早期尝试是 1941 年在美国爱荷华州立大学（Iowa State College）构建的 Atanasoff-Berry 机器。虽然这台机器不可编程且不可靠，但它还是展示出一些现代计算机的概念，并且毫无疑问地推动了当时计算领域的发展。

晶体管计算机的起源同样是一个令人困惑的领域。美国贝尔实验室于 1948 年发明了晶体管，因为功耗低、尺寸小，它很适合构建计算机（虽然早期晶体管的可靠性低于电子管[⊖]）。有一些文字记载误将美国麻省理工学院于 1956 年制造的 TX-0 作为第一台晶体管计算机，实际上曼彻斯特大学于 1953 年投入运行的晶体管计算机才应该享有“第一”的荣誉。

到目前为止，提到的大多数机器都是通过拨动开关或手动插拔电线来“编程”的。从这种意义上说，它们是可编程的，但是速度很慢。相反，现代计算机将程序存储在内存中，这可能更易于编程和调试。第一台存储程序的计算机是曼彻斯特大学构建的小型试验机器（SSEM，或爱称“Baby”）。它于 1948 年首次成功地运行了所存储的程序。另一个早期的存储程序计算机是 Maurice Wilkes 的 EDSAC（Electronic Delay Storage Automatic Calculator），于 1949 年 5 月在剑桥大学开始运行。同样著名的还有美国陆军的 EDVAC（Electronic Discrete Variable Automatic Computer），它也是二进制存储程序计算机，始建于 1944 年，但直到 1951 ~ 1952 年才开始运行。

显然，早期的计算机要么是大学制造的，要么在某种程度上与军事计算有关（例如洛仑兹密码机和 EDVAC）。曼彻斯特大学是一个非常重要但是很低调的角色，被很多计算机历史学家忽视了。在前者中，曼彻斯特大学于 1951 年还设计出世界上第一台商用计算机 Ferranti Mark 1，紧随其后的是由 EDSAC 衍生的 LEO。这台计算机从 1951 年春天开始为 Lyons Tea Houses 运行财务程序。尽管早期计算机业务是以英国为中心，但在接下来的半个世纪里，最大和最重要的商业计算机企业坐落在美国。这种情况一直持续到嵌入式系统出现。随着剑桥开发的 ARM 处理器的广泛使用，至少在一段时间内，这种发展把钟摆摆回了英国。而在 2016 年 9 月，ARM 被卖给了日本 SoftBank 公司。

表 1-1 列出了计算机领域的世界第一，同时给出了它们实际开始运行的年代。

表 1-1 计算机发展史上重要的计算机

年份	研发地点	名称	首次产品
1834	剑桥大学	差分机	可编程计算机
1943	Bletchley	Colossus	电子计算机
1948	曼彻斯特大学	SSEM (Baby)	存储程序计算机
1951	麻省理工学院	Whirlwind 1	实时 I/O 计算机
1953	曼彻斯特大学	晶体管计算机	晶体管计算机
1971	加利福尼亚	Intel 4004 CPU	大众市场 CPU & IC
1979	剑桥大学	Sinclair ZX-79	大众市场家用计算机
1981	纽约	IBM PC	个人计算机
1987	剑桥大学	基于 ARM 的 Acorn A400 系列	拥有 RISC CPU 的家用户计算机
1990	纽约	IBM RS6000 CPU	超标量 RISC 处理器
1998	加利福尼亚	Sun picoJAVA CPU	基于一种语言的计算机

⊖ 在局部真空中包含微小灯丝电极的玻璃热离子阀门是大部分早期计算机的基本逻辑开关。在北美称这种阀门为“真空管”或简单称为“管”。有趣的是，虽然它们现在已不再参与计算，但在很高端的音频放大设备中却是抢手货。