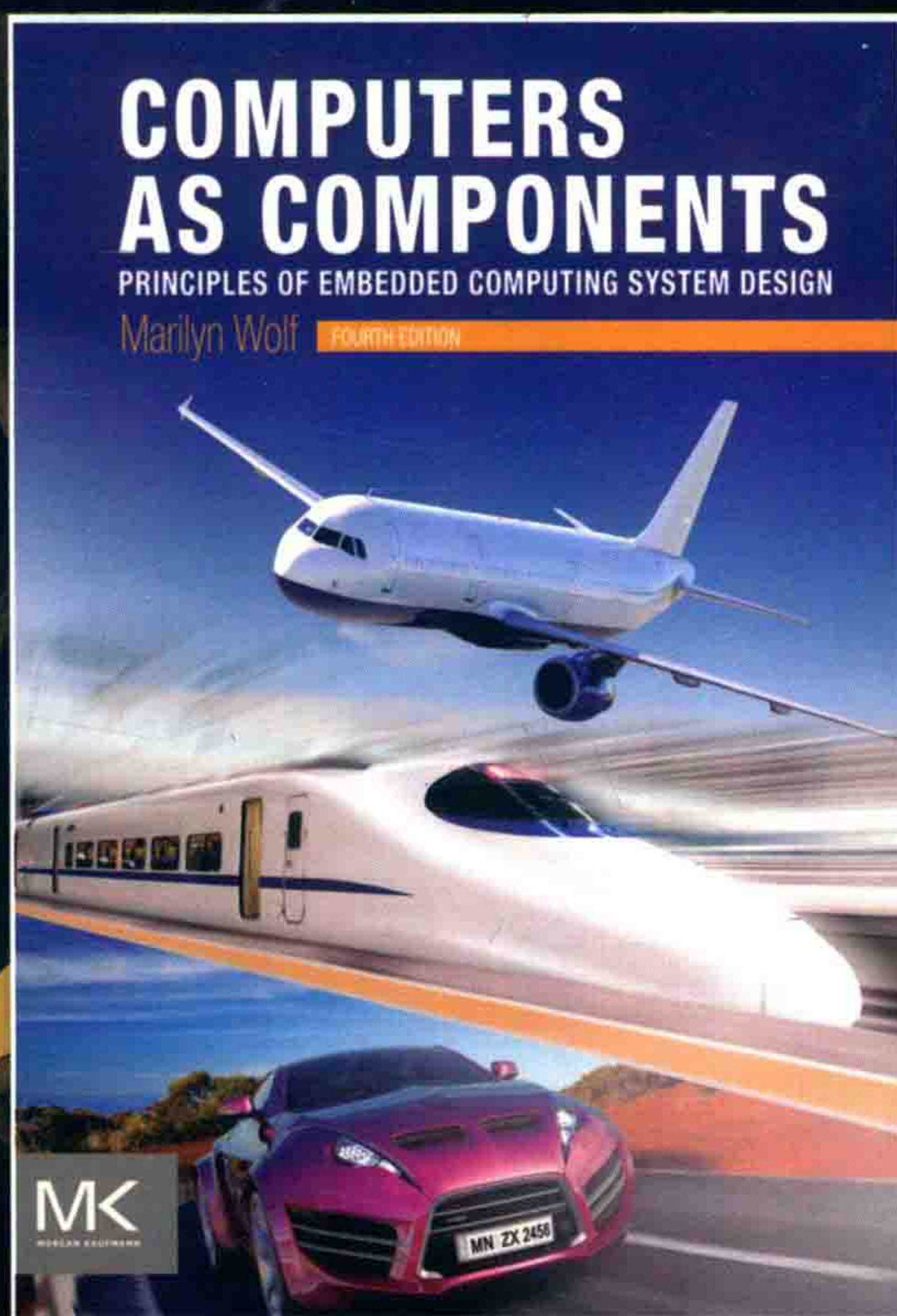


# 嵌入式计算系统设计原理

[美] 玛丽琳·沃尔夫 (Marilyn Wolf) 著 宫晓利 谢彦苗 张金 译  
佐治亚理工学院 南开大学

Computers as Components

Principles of Embedded Computing System Design Fourth Edition



算 机 科 学 丛 书

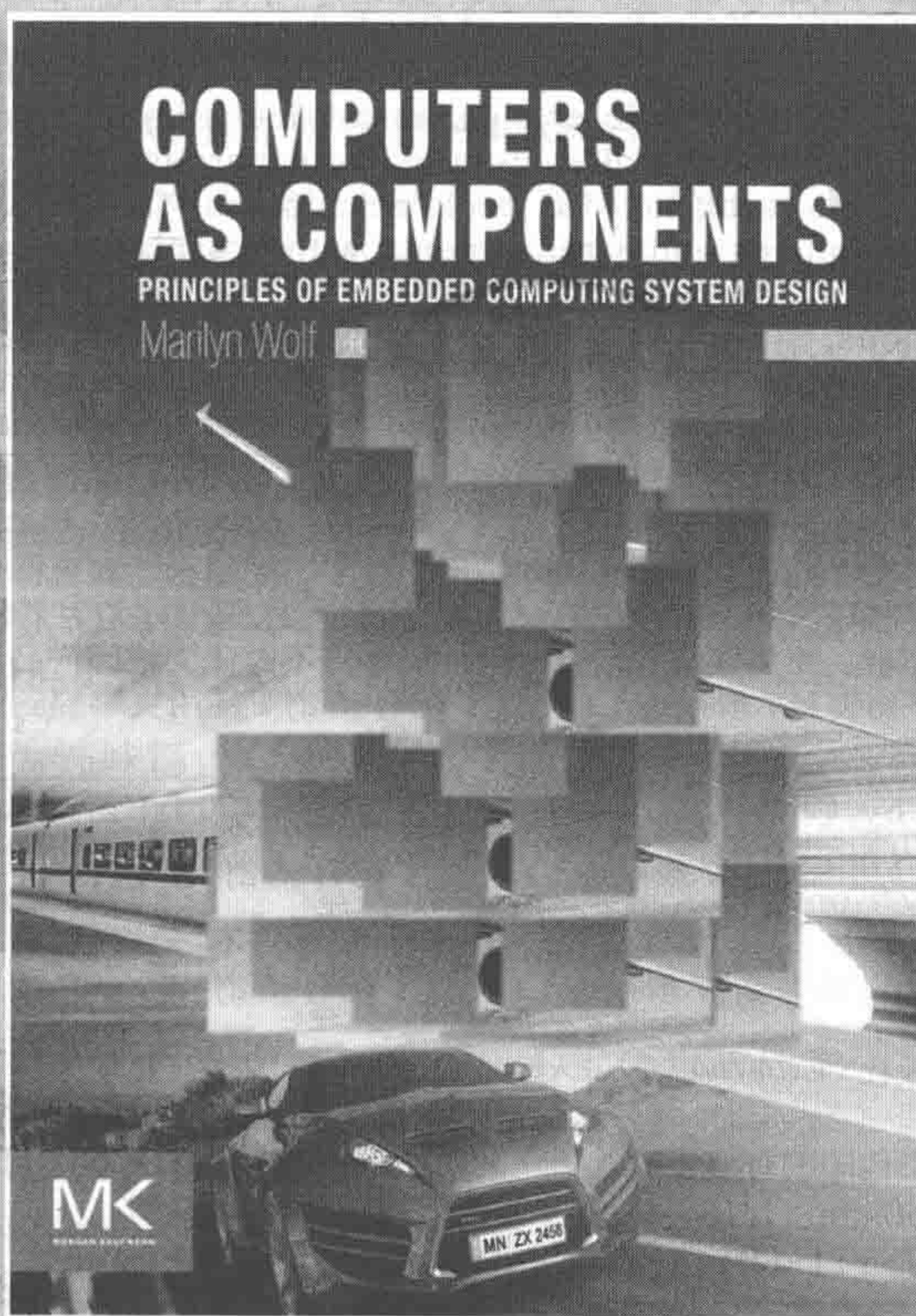
原书第4版

# 嵌入式计算系统设计原理

[美] 玛里琳·沃尔夫 (Marilyn Wolf) 著 宫晓利 谢彦苗 张金 译  
佐治亚理工学院 南开大学

Computers as Components

Principles of Embedded Computing System Design Fourth Edition



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

嵌入式计算系统设计原理 (原书第 4 版)/(美) 玛里琳·沃尔夫 (Marilyn Wolf) 著; 宫晓利, 谢彦苗, 张金译. —北京: 机械工业出版社, 2018.6

(计算机科学丛书)

书名原文: Computers as Components: Principles of Embedded Computing System Design, Fourth Edition

ISBN 978-7-111-60148-7

I. 嵌… II. ①玛… ②宫… ③谢… ④张… III. 微型计算机—系统设计 IV. TP36

中国版本图书馆 CIP 数据核字 (2018) 第 125795 号

本书版权登记号: 图字 01-2017-0509

ELSEVIER

Elsevier (Singapore) Pte Ltd.

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

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

Computers as Components: Principles of Embedded Computing System Design, Fourth Edition  
Marilyn Wolf

Copyright © 2017, 2012, 2008 Elsevier Inc. All rights reserved.

ISBN-13: 978-0-12-805387-4

This translation of Computers as Components: Principles of Embedded Computing System Design, Fourth Edition by Marilyn Wolf was undertaken by China Machine Press and is published by arrangement with Elsevier (Singapore) Pte Ltd.

Computers as Components: Principles of Embedded Computing System Design, Fourth Edition by Marilyn Wolf 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。

《嵌入式计算系统设计原理》(原书第 4 版)(宫晓利 谢彦苗 张金 译)

ISBN: 978-7-111-60148-7

Copyright © 2018 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 in writing 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](http://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: 冯秀泳

印刷: 北京市荣盛彩色印刷有限公司

开本: 185mm×260mm 1/16

书号: ISBN 978-7-111-60148-7

责任校对: 殷虹

版次: 2018 年 7 月第 1 版第 1 次印刷

印张: 25.75

定价: 99.00 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

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: [hzjsj@hzbook.com](mailto: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



## 译者序

Computers as Components: Principles of Embedded Computing System Design, Fourth Edition

从承载人类梦想刺破天际的嫦娥号，漫步火星的太空车，在光滑如镜的冰面上穿梭自如的军用机器人，到时刻伴随身边的智能手机和自动变频的智能空调，嵌入式系统似乎无处不在。

但嵌入式系统的准确界定是什么？这却又是同时困扰初学者和很多业内资深人士的共同问题。嵌入式系统是一个复杂的概念，涉及硬件、指令、编程、操作系统、人机交互等诸多方面的知识。如果以概念为主，容易过于抽象而使得初学者难于理解；如果以应用案例为主，又会陷入到某一个场景之中失之偏颇。

嵌入式系统领域的资深教授 Marilyn Wolf 则以嵌入式系统的设计方法和过程为主线，讲解嵌入式系统每个重要环节的知识，并辅以针对性的案例分析以便读者理解。同时交织穿插讲解嵌入式系统所关注的性能、能耗、可靠性等关键问题，将原本零散、晦涩的嵌入式知识完整地构建成了一个易于学习的体系。不仅于此，作者在本书中还扩展到了物联网、嵌入式多核等新兴知识，极大丰富了嵌入式领域的内涵和外延。因此，我们选择本书进行翻译，希望能在国内嵌入式系统的相关教材中增加一部重量级的著作。

原书作者经验丰富，高屋建瓴，导致本书论述过程中偶有跳跃，为便于读者阅读，我们在翻译过程中增加了一些译者注以使得思路更加连贯，但因水平有限，难免出现画蛇添足的情况，如有不妥之处，请读者不吝赐教。

本书能够顺利成稿，要衷心感谢南开大学嵌入式系统与信息安全实验室的魏鑫、杨皓翔、刘未来、李雨轩、冯帆、孙承君、张瑞等同学在全书的翻译过程中付出的辛苦努力。此外，还要感谢南开大学计算机与控制工程学院的各位老师在本书翻译过程中提供的指导和支持。

本书已更新至第4版，前几版均由国内的同行前辈进行翻译，使得我们在此版翻译的过程中受益良多。限于译者的水平和经验，译文中难免存在不当之处，恳请读者见谅并提出宝贵意见。

译者

2018年3月于南开园

数字系统设计已经进入新的时代。当微处理器设计转向一种典型的优化工作时，只把微处理器作为其部件的嵌入式计算系统设计已经成为一个广阔的新领域。无线系统、可穿戴系统、网络系统、智能家电、工业过程系统、先进汽车系统以及生物接口系统与这个新领域均有交叉，是嵌入式系统设计的热门应用方向。

受传感器、转换器、微电子学、处理器性能、操作系统、通信技术、用户接口和封装技术进步的推动，以及对于用户需要和市场潜能的深入理解，正在涌现出大量新的系统和应用。现在系统设计师和嵌入式系统设计人员的职责就是把这些可能变成现实。

然而现在，嵌入式系统设计仍处于手工阶段。虽然有关硬件组件和软件子系统的知识是很清楚的，但是还没有通用的统筹整个设计过程的系统设计方法，在大多数情况下，仍然是为每个项目完成一个专门的设计。

嵌入式系统设计所面临的某些挑战源于基础技术的改变以及系统各部件全部正确地混合和集成在一起的奥妙，另一些挑战源于新的并且常常是不熟悉的系统需求。此外，通信和协作基础设施及技术的改善使得设计对市场需求的响应达到了前所未有的速度。但是，还没有有效的设计方法和相关的设计工具足以迅速应对这些挑战。

在 VLSI 时代的开始阶段，晶体管和导线是基本部件，快速设计单芯片计算机是一种梦想。今天，CPU 和各种专用处理器以及子系统仅是一些基本部件，但快速、高效地设计极复杂的嵌入式系统却仍是一个梦想。现在不但系统规格说明极其复杂，而且面临着实时期限、低功耗、有效地支持复杂实时用户界面、强劲的成本竞争以及设计的系统必须可升级等问题。

Wayne Wolf<sup>⊖</sup>教授编写了系统地处理大量新的系统设计需求和挑战的第一本教科书。Wolf 提出了嵌入式系统设计的形式体系和方法学，可以为嵌入式系统架构师提供帮助。这些架构师要真正理解各种组件的技术原理和系统设计基础，并运用书中的方法构造出新型的“超薄”系统。

从分析每一种技术基础出发，Wolf 教授为规范和建造系统结构及行为提供了形式化方法，然后通过一系列示例解析这些思想，而且还仔细研究了所涉及的复杂性以及如何系统地处理这些复杂问题。你会提前清楚地理解这些设计问题的本质，并知道攻克这些难关的关键方法和工具。

作为嵌入式系统设计的第一本教科书，本书必将成为在这个重要和最新出现的领域里获得知识的无价工具。本书亦可以作为实际设计工作的可靠指南。我向读者强烈推荐本书。

Lynn Conway

密歇根大学电子工程和计算机科学系名誉教授

⊖ Wayne Wolf 是本书作者 Marilyn Wolf 的曾用名。——编辑注

## 第4版前言

Computers as Components: Principles of Embedded Computing System Design, Fourth Edition

准备本书第4版的过程使我意识到我的年龄有多大。在1999年年底，我完成了第1版的最终草稿。从那时起，嵌入式计算技术发展迅猛，但是核心原理依然基本保持不变。我对整本书进行了修改：整理问题，修改文稿，对某些章节重新排序以改进思路，并删除了一些不重要的内容。

本书在两方面做出了巨大改动。第一，新增了物联网（IoT）这一章；第二，将安全性贯穿全书。自第3版出版以来，IoT就已作为一个重要主题出现，但是它的实现依赖于现存的技术和知识。在第4版中，新的IoT章节综述了几个用于IoT应用的无线网络，也给出了一些IoT系统组织的模型。安全性对于嵌入式系统一直都很重要，虽然本书第1版就已讨论过医疗设备的安全性，但是一系列事件又突出了这一主题的关键特性。

在前几版中，一些高级的知识点都放在第8章，包括多处理器片上系统和网络嵌入式系统等。第4版将这些内容扩展，并分为三章：第8章涵盖OSI和网络协议，以及IoT特定主题的知识；第9章探讨汽车环境中的网络嵌入式系统，并涵盖安全性方面的几个例子；第10章讲述多处理器片上系统及其应用。

和前几版一样，在本书网站 <http://www.marilynwolf.us> 可以找到所有相关内容。一些外部Web资源的链接也在该网站上。此外，我的新博客 <http://embeddedcps.blogspot.com/> 提供了一些嵌入式计算人员感兴趣的话题。

我要感谢编辑 Nate McFadden 给予的帮助和指导。当然，本书中的任何问题都是我的责任。

Marilyn Wolf

本书第3版反映了我对嵌入式计算的深入思考以及对本书读者的若干建议，其中一个重要的目标是扩大嵌入式计算应用的范围。学习有关数码相机和汽车之类的主题需要付出很多努力。我希望本书可以为这些系统中直接影响嵌入式计算设计者决策的部分提供一些有用的见解。我也扩大了示例处理器的范围，包括尖端的处理器，如 TI C64x 和高级 ARM 扩展 (advanced ARM extension)，同时还包括 PIC16F，并通过它描述小型 RISC 嵌入式处理器的特性。最后重新组织了网络和多处理器章节，使这些紧密相关的主题看起来更统一。读者可以在课程网站 <http://www.marilynwolf.us> 寻找附加的材料，这个网站包含了上面所说的所有内容，还有实验样例以及获取附加信息的提示。

我要感谢 Nate McFadden、Todd Green 和 Andre Cuello 的耐心编辑以及在本书修订过程中对我的关心。我还要感谢匿名评论者和科罗拉多大学的 Andrew Pleszkun 教授对本书草稿的中肯建议。特别感谢 David Anderson、Phil Koopman 和 Bruce Jacob 帮助我理解了一些内容。同样要感谢 2011 年的亚特兰大“末日暴雪”给了我大量不受打扰的写作时间。

最重要的是，这是我感谢父亲的最好时机。他教会我如何工作：不仅仅是教会我怎样做具体的事情，更重要的是教会我如何处理问题，开拓思路，然后把它们转化为成果。一直以来，他教我如何体贴关心他人。感谢您，父亲。

Marilyn Wolf



## 第 2 版前言

Computers as Components: Principles of Embedded Computing System Design, Fourth Edition

相对 2000 年本书第 1 版出版的时候，如今嵌入式计算更为重要。更多的产品中使用了嵌入式处理器，从玩具到飞机都有应用。片上系统现在使用几百个 CPU。手机朝着新的标准计算平台方向发展。就像 2006 年 9 月《IEEE Computer》杂志上我的专栏中所指出的那样，当今世界至少有 50 万嵌入式系统程序员，可能接近 80 万。

在这一版中，我尽力做了更新和补充。本书的一个主要改变是使用 TI C55x DSP。我慎重地重写了关于实时调度的讨论。尝试将性能分析主题尽可能在更多的抽象层次上扩展。指出多处理器在甚至最平凡的嵌入式系统中的重要性，此版对硬 / 软件协同设计和多处理器也进行了更通用的介绍。

计算机教学领域的一个改变是，本教材成为越来越低年级的课本。过去用于研究生的教材现在用于高年级本科生；在可预见的未来，本书的部分内容将可作为大学二年级教材。我认为可以选取本书的部分内容去覆盖更先进和更基础的课程。一些高年级学生可能不需要前面章节的背景知识，这样可以把更多时间花在软件性能分析、调度和多处理器上。当开设介绍性课程时，软件性能分析可作为探索微处理器体系结构和软件体系结构的一个可选方案；这样的课程可以关注前几章的内容。

本书和我的其他书的新网站是 <http://www.waynewolf.us>。在这个网站里，可以找到本书相关材料的概括、实验建议，还可找到关于嵌入式系统的更多信息的网站链接。

### 致谢

感谢许多帮助我完成第 2 版的人。德州仪器的 Cathy Wicks 和 Naser Salameh 在理解 C55x 上给了我非常有价值的帮助。FreeRTOS.org 的 Richard Barry 不仅慷慨地允许我引用其操作系统的源码，还帮我澄清代码的解释。本书的编辑是 Morgan Kaufmann 出版社的 Chuck Glaser，他知道何时需要耐心，何时需要鼓励，何时需要诱导。当然，还要感谢 Nancy 和 Alec 耐心为我录入。本书的任何问题，不管是大小，自然都是我个人的责任。

Marilyn Wolf

微处理器早已成为我们生活的一部分，然而，微处理器足够强大到能执行真正复杂的功能还是近几年的事。在摩尔定律的驱动下，微处理器飞速发展，同时促使嵌入式计算成为一门学科。在微处理器的早期阶段，所有组件相对较小也较简单，需要且期望把一些单独的指令和逻辑门集中在一起。今天，当系统包含了几千万个晶体管和数万行高级语言代码时，我们必须使用有助于处理复杂性的设计技术。

本书试图捕捉嵌入式计算这一新学科的某些基本原理和技术。嵌入式计算所面临的一些挑战在台式机计算世界中是众所周知的。例如，为从带流水线的高速缓存体系结构中获得最高性能，经常需要仔细分析程序轨迹。类似地，随着嵌入式系统的复杂性不断增加，在软件工程中针对特定复杂系统开发的技术变得十分重要。另外一个例子是设计多进程系统。对于台式机上使用的通用操作系统和嵌入式系统使用的实时操作系统，二者的需求是截然不同的。过去 30 年针对大型实时系统开发的实时技术如今已普遍应用于基于微处理器的嵌入式系统中。

嵌入式计算还面临一些新的挑战。一个较好的例子是功耗问题。在传统计算机系统中功耗已经不是一个主要考虑因素，但是对于用电池供电的嵌入式计算机，这是一个基本考虑因素，而且在功耗容量受重量、成本或噪声等限制的情况下是十分重要的。另外一个挑战是截止时限驱动的程序设计。嵌入式计算机常常对程序完成的期限做硬性限制，这种形式的限制在台式机世界里是罕见的。随着嵌入式处理器越来越快，高速缓存和其他 CPU 单元也使得执行时间越来越难以预测。然而，通过仔细分析和巧妙编程，即使面对高速缓存等不可预测的系统组件，我们也可以设计出可预测执行时间的嵌入式程序。

幸运的是，有许多工具可用来处理复杂嵌入式系统所面临的挑战：高级语言、程序性能分析工具、进程和实时操作系统，等等。但是理解这些工具如何协调地一起工作，这本身就是一项很复杂的任务。本书提供了一种自底向上的方法来理解嵌入式系统设计技术。通过先理解微处理器硬件和软件的基础知识，我们就能获得有助于创建复杂系统的强有力的抽象能力。

## 写给嵌入式系统专业人员

本书不是一本用来理解某种特定微处理器的手册。为什么在这里呈现的技术你会感兴趣呢？有两个理由。第一，诸如高级语言编程和实时操作系统这样的技术对于构造实际的大型复杂嵌入式系统是非常重要的。生产会因为不能工作的错误系统设计而被弄得杂乱无章，系统之所以不能工作是因为它们的设计者试图从出现的问题中寻求解决方法，而不是从问题中走出来换个更大的视角研究问题。第二，用于构造嵌入式系统的组件是经常变化的，但其原理不变。一旦你掌握了创建复杂嵌入式系统所涉及的基本原理，就可以迅速地学习一种新的微处理器（或编程语言），并且把同样的基本原理用于新的组件。

## 写给教师

传统的微处理器系统设计起源于 20 世纪 70 年代，当时微处理器的种类相对有限。传统课程强调定制硬件和软件来构建一个完整系统。因此，它只强调某一特定微处理器的特性，

包括其指令系统、总线接口，等等。

本书采用更抽象的途径研究嵌入式系统。书中利用一切机会来讨论实际组件和应用，但本质上不是一本微处理器数据手册，因此它的论述方法初看起来是新奇的。本书不专注于某种特定类型的微处理器，而是试图研究更通用的例子，以得出更普遍适用的原理。我认为这种方法更有利于教学，而从长远角度来看对于学生也更有用。对于教学更有利是因为不必太过于依赖复杂的实验室装置，而只需花费一些时间在纸上练习，进行模拟和编程练习。对于学生更有用是因为他们在这一领域最终工作时所使用的组件和设施与学校的肯定是不同的，一旦学生掌握了基础知识，他们学习新组件的细节就会容易得多。

对于获得有关嵌入式系统的物理直觉，实践经验特别重要。某些硬件设计经验是非常宝贵的，我认为每一个学生都应该知道燃烧塑封集成电路包装盒的气味。但我强烈建议你避免专注于硬件设计。如果花费太多的时间去构建硬件平台，你将没有足够的时间去编写有趣的程序。一个实际问题是，大多数课程没有时间让学生用高性能 I/O 设备和可能的多处理器来建造复杂的硬件平台。多数学生可以通过测量和评价一个现有的硬件平台来学习硬件知识。编制复杂嵌入式系统程序的实践也可以教给学生相当多的硬件知识，调试中断驱动代码是学生基本不会忘记的一种经验。

本书的主页 ([www.mkp.com/embed](http://www.mkp.com/embed)) 中包括本书相关材料的概括、教师手册<sup>⊖</sup>、实验材料、到相关 Web 站点的链接，以及包含习题解答的受密码保护的 ftp 站点的链接。

## 致谢

感谢许多帮助我准备这本书的人。一些人给了我关于本书各个方面的建议：关于规格说明的 Steve Johnson (印第安纳大学)，关于程序跟踪的 Louise Trevillyan 和 Mark Charney (均在 IBM 研究院)，关于高速缓存失效方程的 Margaret Martonosi (普林斯顿大学)，关于低功耗的 Randy Harr (美国新思科技公司)，关于分布式系统的 Phil Koopman (卡内基 - 梅隆大学)，关于低功耗计算与累加器的 Joerg Henkel (NEC C&C 实验室)，关于实时操作系统的 Lui Sha (伊利诺伊大学)，关于 ARM 体系结构的 John Rayfield (ARM 公司)，关于编译器和 SHARC 的 David Levine (美国模拟器件公司)，以及关于 SHARC 的 Con Korikis (美国模拟器件公司)。许多人员在各阶段对本书进行了审阅：David Harris (哈维姆德学院)，Jan Rabaey (加州大学伯克利分校)，David Nagle (卡内基 - 梅隆大学)，Randy Harr (美国新思科技公司)，Rajesh Gupta、Nikil Dutt、Frederic Doucet 和 Vivek Sinha (加州大学欧文分校)，Ronald D. Williams (弗吉尼亚大学)，Steve Sapiro (SC 协会)，Paul Chow (多伦多大学)，Bernd G. Wenzel (Eurostep)，Steve Johnson (印第安纳大学)，H. Alan Mantooth (阿肯色大学)，Margarida Jacome (得克萨斯大学奥斯汀分校)，John Rayfield (ARM 公司)，David Levine (美国模拟器件公司)，Ardsher Ahmed (马萨诸塞大学 / 达特茅斯大学)，Vijay Madisetti (佐治亚理工学院)。还要特别感谢编辑 Denise Penrose，Denise 费了很大精力寻找本书的潜在用户并和他们交流，帮助我们了解读者想要学什么。特别感谢她的洞察力和坚持。Cheri Palmer 和她的出版团队在令人难以置信的紧迫日程内出色地完成了工作。当然，所有的错误和失误都是我的。

Wayne Wolf

⊖ 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，需要的教师访问爱思唯尔的教材网站 <https://textbooks.elsevier.com/> 进行申请。——编辑注

出版者的话	
译者序	
第1版序言	
第4版前言	
第3版前言	
第2版前言	
第1版前言	
<b>第1章 嵌入式计算</b>	<b>1</b>
1.1 引言	1
1.2 复杂系统与微处理器	1
1.2.1 嵌入式计算机	2
1.2.2 嵌入式计算应用的特点	3
1.2.3 为什么使用微处理器	4
1.2.4 信息物理系统	6
1.2.5 防危性和安全性	6
1.2.6 嵌入式计算系统设计所面临的挑战	8
1.2.7 嵌入式计算系统的性能	9
1.3 嵌入式系统设计过程	9
1.3.1 需求	10
1.3.2 规格说明	14
1.3.3 体系结构设计	14
1.3.4 设计硬件与软件组件	16
1.3.5 系统集成	16
1.3.6 系统设计的形式化方法	16
1.3.7 结构描述	17
1.3.8 行为描述	20
1.4 设计示例: 模型火车控制器	22
1.4.1 需求	23
1.4.2 DCC	24
1.4.3 概念性规格说明	25
1.4.4 详细规格说明	27
1.4.5 经验总结	32
1.5 本书导读	32
1.5.1 第2章: 指令集	33
1.5.2 第3章: CPU	33
1.5.3 第4章: 计算平台	33
1.5.4 第5章: 程序设计与分析	34
1.5.5 第6章: 进程和操作系统	34
1.5.6 第7章: 系统设计技术	35
1.5.7 第8章: 物联网系统	35
1.5.8 第9章: 汽车和航天系统	35
1.5.9 第10章: 嵌入式多处理器	36
1.6 总结	36
我们学到了什么	36
扩展阅读	36
问题	37
上机练习	38
<b>第2章 指令集</b>	<b>39</b>
2.1 引言	39
2.2 预备知识	39
2.2.1 计算机体系结构分类	39
2.2.2 汇编语言	41
2.2.3 VLIW 处理器	42
2.3 ARM 处理器	43
2.3.1 处理器和存储体系	44
2.3.2 数据运算	44
2.3.3 控制流程	49
2.3.4 ARM 的高级特性	54
2.4 PICmicro 系列中端微处理器	55
2.4.1 处理器和存储体系	55
2.4.2 数据操作	55
2.4.3 控制流程	57
2.5 TI C55x DSP	58
2.5.1 处理器和存储体系	58
2.5.2 寻址模式	61
2.5.3 数据操作	62
2.5.4 控制流程	63
2.5.5 C 语言编程指南	64
2.6 TI C64x	65
2.7 总结	68

我们学到了什么 .....	68	4.2 基本的计算平台 .....	116
扩展阅读 .....	68	4.2.1 平台硬件组件 .....	116
问题 .....	68	4.2.2 平台软件组件 .....	118
上机练习 .....	69	4.3 CPU 总线 .....	119
<b>第 3 章 CPU</b> .....	<b>70</b>	4.3.1 总线结构和协议 .....	119
3.1 引言 .....	70	4.3.2 直接内存访问 .....	124
3.2 输入 / 输出编程 .....	70	4.3.3 系统总线配置 .....	125
3.2.1 输入 / 输出设备 .....	70	4.4 存储设备和系统 .....	127
3.2.2 输入 / 输出原语 .....	72	4.5 基于计算平台的系统设计 .....	129
3.2.3 忙等待 I/O .....	73	4.5.1 示例平台 .....	129
3.2.4 中断 .....	74	4.5.2 选择平台 .....	129
3.3 特权模式、异常和陷阱 .....	85	4.5.3 知识产权 .....	131
3.3.1 特权模式 .....	85	4.5.4 开发环境 .....	132
3.3.2 异常 .....	85	4.5.5 看门狗定时器 .....	132
3.3.3 陷阱 .....	86	4.5.6 调试技术 .....	133
3.4 协处理器 .....	86	4.5.7 调试中的困难和挑战 .....	135
3.5 存储系统机制 .....	86	4.6 消费类电子设备的体系结构 .....	136
3.5.1 高速缓存 .....	87	4.6.1 消费类电子设备的用例和	
3.5.2 存储管理单元和地址转换 .....	91	需求 .....	136
3.6 CPU 性能 .....	94	4.6.2 文件系统 .....	137
3.6.1 流水线技术 .....	94	4.7 平台级性能分析 .....	138
3.6.2 高速缓存的性能 .....	98	4.8 平台级电源管理 .....	141
3.7 CPU 的功耗 .....	99	4.9 设计示例：闹钟 .....	142
3.7.1 CMOS 功耗 .....	99	4.9.1 需求 .....	142
3.7.2 电源管理模式 .....	99	4.9.2 规格说明 .....	143
3.7.3 程序级电源管理 .....	101	4.9.3 系统体系结构 .....	146
3.8 防危性和安全性 .....	102	4.9.4 组件设计和测试 .....	147
3.9 设计示例：数据压缩器 .....	103	4.9.5 系统集成和测试 .....	147
3.9.1 需求和算法 .....	103	4.10 设计示例：音频播放器 .....	147
3.9.2 规格说明 .....	105	4.10.1 工作原理和需求 .....	147
3.9.3 程序设计 .....	106	4.10.2 规格说明 .....	149
3.9.4 测试 .....	111	4.10.3 系统体系结构 .....	150
3.10 总结 .....	112	4.10.4 组件设计和测试 .....	151
我们学到了什么 .....	112	4.10.5 系统集成和调试 .....	151
扩展阅读 .....	112	4.11 总结 .....	151
问题 .....	112	我们学到了什么 .....	151
上机练习 .....	115	扩展阅读 .....	151
<b>第 4 章 计算平台</b> .....	<b>116</b>	问题 .....	152
4.1 引言 .....	116	上机练习 .....	154

<b>第 5 章 程序设计与分析</b> .....	155	5.13.2 规格说明 .....	214
5.1 引言 .....	155	5.13.3 系统体系结构 .....	216
5.2 嵌入式程序的组件 .....	155	5.13.4 组件设计和测试 .....	218
5.2.1 状态机 .....	155	5.13.5 系统集成和测试 .....	218
5.2.2 循环缓冲区和面向流的 程序设计 .....	157	5.14 总结 .....	218
5.2.3 队列与生产者 / 消费者系统 .....	161	我们学到了什么 .....	218
5.3 程序模型 .....	163	扩展阅读 .....	218
5.3.1 数据流图 .....	163	问题 .....	219
5.3.2 控制 / 数据流图 .....	164	上机练习 .....	225
5.4 汇编、链接和加载 .....	166	<b>第 6 章 进程和操作系统</b> .....	227
5.4.1 汇编器 .....	167	6.1 引言 .....	227
5.4.2 链接 .....	170	6.2 多任务和多进程 .....	227
5.4.3 目标代码设计 .....	171	6.3 多速率系统 .....	229
5.5 编译技术 .....	172	6.3.1 进程的时间约束 .....	230
5.5.1 编译过程 .....	172	6.3.2 CPU 使用效率度量标准 .....	234
5.5.2 基本编译方法 .....	173	6.3.3 进程状态和调度 .....	234
5.5.3 编译器优化方法 .....	179	6.3.4 运行周期性进程 .....	235
5.6 程序级性能分析 .....	185	6.4 抢占式实时操作系统 .....	237
5.6.1 程序性能分析 .....	186	6.4.1 两个基本概念 .....	237
5.6.2 测量驱动的性能分析 .....	190	6.4.2 进程和上下文 .....	238
5.7 软件性能优化 .....	193	6.4.3 进程和面向对象设计 .....	240
5.7.1 循环的基本优化 .....	193	6.5 基于优先级的调度 .....	241
5.7.2 针对高速缓存的优化 .....	194	6.5.1 单调速率调度 .....	242
5.7.3 性能优化策略 .....	196	6.5.2 最早截止时间优先调度 .....	244
5.8 程序级的能量和功率分析及优化 .....	197	6.5.3 RMS 与 EDF 比较 .....	248
5.9 程序长度的分析和优化 .....	199	6.5.4 共享资源 .....	248
5.10 程序验证和测试 .....	200	6.5.5 优先级反转 .....	250
5.10.1 白盒测试 .....	201	6.5.6 低功耗调度 .....	250
5.10.2 黑盒测试 .....	205	6.5.7 对模型假设的进一步分析 .....	250
5.10.3 功能性测试 .....	206	6.6 进程间通信机制 .....	252
5.11 防危性与安全性 .....	207	6.6.1 共享内存通信 .....	252
5.12 设计示例: 软件调制解调器 .....	207	6.6.2 消息传递 .....	253
5.12.1 工作原理和需求 .....	207	6.6.3 信号 .....	254
5.12.2 规格说明 .....	209	6.6.4 信箱 .....	254
5.12.3 系统体系结构 .....	209	6.7 评估操作系统性能 .....	255
5.12.4 组件设计和测试 .....	210	6.8 实时操作系统示例 .....	258
5.12.5 系统集成和测试 .....	210	6.9 设计示例: 电话答录机 .....	263
5.13 设计示例: 数码相机 .....	210	6.9.1 工作原理和需求 .....	263
5.13.1 工作原理和需求 .....	210	6.9.2 规格说明 .....	265

6.9.3 系统体系结构	267	8.2 IoT 系统应用	303
6.9.4 组件设计和测试	268	8.3 IoT 系统体系结构	304
6.9.5 系统集成和测试	269	8.4 IoT 网络	305
6.10 设计示例: 发动机控制单元	269	8.4.1 OSI 模型	306
6.10.1 工作原理和需求	269	8.4.2 互联网协议	306
6.10.2 规格说明	270	8.4.3 IoT 网络的概念	308
6.10.3 系统体系结构	270	8.4.4 蓝牙和低功耗蓝牙	310
6.10.4 组件设计和测试	272	8.4.5 802.15.4 和无线个域网	312
6.10.5 系统集成和测试	272	8.4.6 Wi-Fi	313
6.11 总结	272	8.5 数据库和时间轮	315
我们学到了什么	273	8.5.1 数据库	315
扩展阅读	273	8.5.2 时间轮	317
问题	273	8.6 示例: 智能家居	317
上机练习	277	8.7 总结	319
<b>第 7 章 系统设计技术</b>	<b>279</b>	我们学到了什么	319
7.1 引言	279	扩展阅读	320
7.2 设计方法	279	问题	320
7.2.1 为什么需要设计方法	279	上机练习	320
7.2.2 设计流程	281	<b>第 9 章 汽车和航天系统</b>	<b>321</b>
7.3 需求分析	285	9.1 引言	321
7.4 规格说明	286	9.2 汽车和飞机中的网络控制系统	321
7.4.1 面向控制的规格说明语言	286	9.3 车载网络	323
7.4.2 改进的规格说明	288	9.3.1 CAN 总线	323
7.5 系统分析和架构设计	290	9.3.2 其他汽车网络	325
7.6 可靠性、防危性和安全性	293	9.4 防危性和安全性	326
7.6.1 示例	293	9.5 总结	327
7.6.2 质量保证技术	296	我们学到了什么	328
7.6.3 验证规格说明	297	扩展阅读	328
7.6.4 设计审查	298	问题	328
7.6.5 面向安全的方法	299	上机练习	328
7.7 总结	301	<b>第 10 章 嵌入式多处理器</b>	<b>329</b>
我们学到了什么	301	10.1 引言	329
扩展阅读	301	10.2 为什么需要多处理器	329
问题	301	10.3 多处理器的种类	331
上机练习	302	10.4 MPSOC 和共享内存多处理器	332
<b>第 8 章 物联网系统</b>	<b>303</b>	10.4.1 异构共享内存多处理器	333
8.1 引言	303	10.4.2 加速器	333

10.4.3	加速器性能分析	335	10.6	应用示例：光盘	348
10.4.4	调度和分配	337	10.7	总结	351
10.4.5	系统集成	339	我们学到了什么	351	
10.4.6	调试	342	扩展阅读	351	
10.5	设计示例：视频加速器	342	问题	351	
10.5.1	视频压缩	342	上机练习	352	
10.5.2	算法和需求	344	术语表	353	
10.5.3	规格说明	345	参考文献	366	
10.5.4	架构	345	索引	374	
10.5.5	组件设计	347			
10.5.6	系统测试	348			



# 嵌入式计算

## 本章要点

- 我们为什么在系统中嵌入微处理器。
- 嵌入式计算和信息物理系统 (cyber-physical system) 设计的难点与独到之处。
- 设计方法。
- 系统规格说明。
- 本书导读。

## 1.1 引言

本章将为学习设计嵌入式计算系统打下基础。为了理解设计过程，首先我们要理解如何将微处理器用于过程控制、用户界面、信号处理和其他任务，以及为什么要这么做。如今微处理器已经很普遍，以至于我们很容易忘记没有它的时候完成任务有多困难。

首先我们回顾一下微处理器的各种用途。然后回顾微处理器被用于系统设计的主要原因——实现复杂的动作控制，缩短产品的设计周期，增加产品设计的灵活性，等等。接下来的 1.2 节，我们将通过一个系统示例来了解设计系统的主要步骤。1.3 节深入分析了嵌入式系统的设计规范和设计过程中使用到的技术——这些规范和技术将贯穿本书中的所有设计过程。在 1.4 节，我们用一个模型火车控制器作为例子来展示如何应用这些规范和技术。1.5 节对每一章的内容进行了简要介绍。

## 1.2 复杂系统与微处理器

一般我们认为笔记本电脑就是一台计算机，其实它是众多的计算机系统类型中的一员。计算机是一个存储着程序的机器，它能够从内存中获取并执行指令。我们可以给计算机连接不同类型的设备，加载不同类型的软件，也可以基于计算机建立许多不同类型的系统。

那么什么是嵌入式计算机系统呢？不严谨地说，它是任何内部包含了一部可编程计算机的设备，但它本身不是作为通用计算机设计的。因此，个人计算机 (Personal Computer, PC) 本身并不是一个嵌入式计算系统，但使用微处理器构建的各种产品，例如传真机或者时钟，都是嵌入式计算系统。

嵌入式计算系统设计在许多类型的产品设计中都有重要的作用。汽车、手机甚至家用电器中都广泛使用了微处理器。在这些应用领域中，设计师们必须判断出在哪里使用微处理器，然后设计一个硬件平台，连接相应的 I/O (输入/输出) 设备以支持目标任务，最后设计出相应的软件以控制任务处理的过程。这非常类似计算机工程。与机械设计、热力学分析一样，计算机工程也是可以应用于许多不同领域的基础学科。嵌入式计算系统的设计不是孤立的，在设计过程中遇到的许多挑战并不是计算机工程问题——例如，有可能是机械或模拟电路问题。本书中我们主要对嵌入式计算机本身做研究，所以我们将专注于那些与最终的产品功能直接相关的硬件和软件。

1