

原书第2版

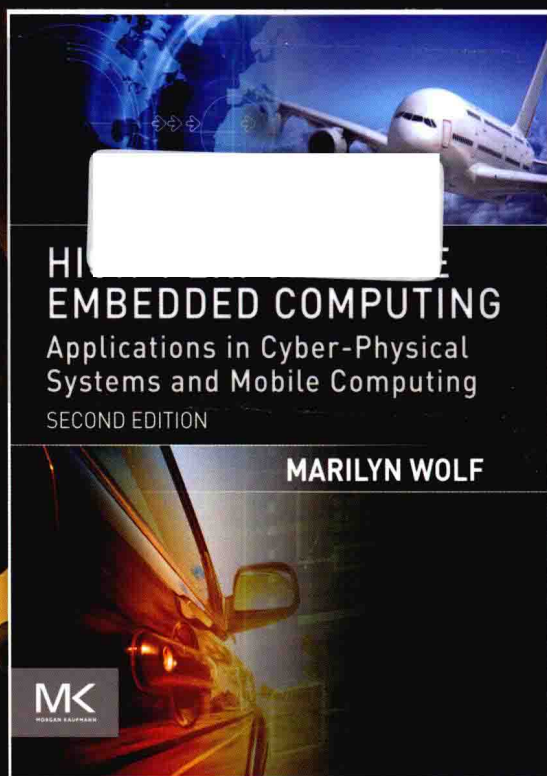
高性能嵌入式计算

[美] 玛里琳·沃尔夫 (Marilyn Wolf) 著

刘彦 付彬 李仁发 译

High-Performance Embedded Computing

Applications in Cyber-Physical Systems and Mobile Computing Second Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

原书第2版

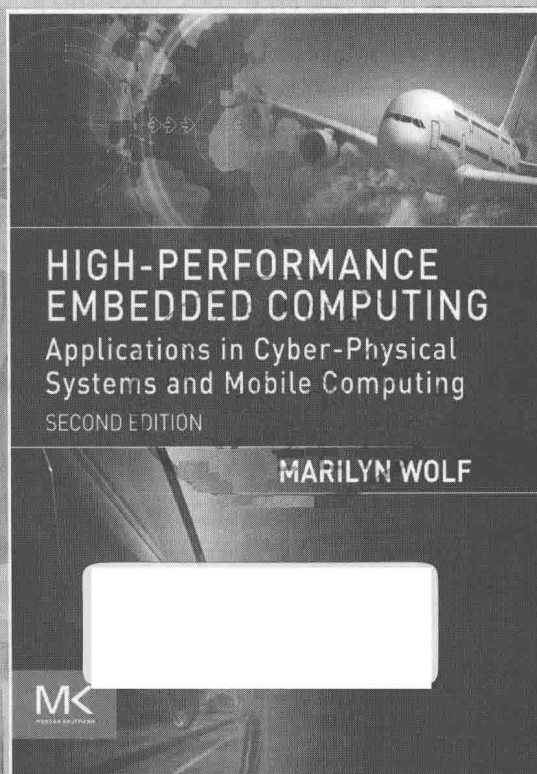
高性能嵌入式计算

[美] 玛里琳·沃尔夫 (Marilyn Wolf) 著

刘彦 付彬 李仁发 译

High-Performance Embedded Computing

Applications in Cyber-Physical Systems and Mobile Computing Second Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

高性能嵌入式计算 (原书第 2 版)/(美) 玛里琳·沃尔夫 (Marilyn Wolf) 著; 刘彦, 付彬, 李仁发译. —北京: 机械工业出版社, 2016.5

(计算机科学丛书)

书名原文: High-Performance Embedded Computing: Applications in Cyber-Physical Systems and Mobile Computing, Second Edition

ISBN 978-7-111-54051-9

I. 高… II. ①玛… ②刘… ③付… ④李… III. 微型计算机—系统设计 IV. TP36

中国版本图书馆 CIP 数据核字 (2016) 第 129649 号

本书版权登记号: 图字: 01-2015-1271

High-Performance Embedded Computing: Applications in Cyber-Physical Systems and Mobile Computing, Second Edition

Marilyn Wolf

ISBN: 978-0-12-410511-9

Copyright © 2014, 2007 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。

本书从性能、功率和能耗以及成本等方面讲述现代嵌入式计算系统的设计方法。第 1 章概述嵌入式计算, 第 2 章介绍各种处理器, 第 3 章讲述编程, 第 4 章讨论进程、操作系统, 第 5 章介绍多处理器体系结构, 第 6 章讲述多处理器软件, 第 7 章介绍系统级设计和软硬件协同设计, 第 8 章讲述物理信息系统的设计。

本书包含大量实例, 涵盖嵌入式计算领域的高级主题, 适合对嵌入式系统感兴趣的专业人员和师生阅读。

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: 谢晓芳

印 刷: 中国电影出版社印刷厂

开 本: 185mm×260mm 1/16

书 号: ISBN 978-7-111-54051-9

责任校对: 殷 钰

版 次: 2016 年 6 月第 1 版第 1 次印刷

印 张: 21.5

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

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

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

电子邮件：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

嵌入式计算系统的发展历史悠久，应用范围特别广泛。本书作者是嵌入式系统领域的著名学者，并具有丰富的产品开发经验。本书针对嵌入式计算所涉及的各个方面进行了框架性的综述，包括嵌入式处理器、编程与调度方法、多核处理器和系统设计方法等，第2版中增加了关于信息物理系统的章节，使其内容更为充实。

尽管嵌入式计算使用的很多技术来源于通用计算系统，但它也为设计者和使用者提出了很多独特的、新的挑战。嵌入式系统的应用通常对计算系统的实时性、功耗或者成本有更为严格的约束，无论是系统设计人员还是应用软件开发人员都需要对系统软硬件性能和应用特征有更为深入的了解，才能开发出满足诸多约束的、具有竞争力的产品。本书对计算机科学与工程相关领域基本概念和技术着墨不多，主要关注与高性能嵌入式计算领域相关的主题及其发展脉络，力图为每个特殊或者独特的课题找到其主要解决方法的重要经典文献，并全面阐述该领域当前最新研究进展。

要做好本书的翻译工作确实不容易。书中不仅涵盖了诸多新技术、新方法和新产品，而且涉及多个学科与领域的专业知识，对于我们来说，翻译本书也是一个不断学习、不断改进的过程。书中难免有疏漏之处，我们表示深深的歉意并恳请读者批评指正。

译者

在第2版中，我试图用新的眼光来看待嵌入式计算和可以用于设计高性能嵌入式系统的各种技术。当前各种应用使用嵌入式多处理器和复杂的软件栈：智能手机、飞机、汽车、电力设备等。这些复杂的应用需要复杂的硬件、软件和设计方法。

我更新了每一章，部分章节进行了较多的修改。新增关于信息物理系统（CPS）的一章（第8章）。CPS在本书第1版出版后作为控制理论和嵌入式计算的一种综合体出现。虽然CPS影响了本书其他章节所描述的许多技术，但许多新的结果又应用在CPS的设计之中。我试图概述这些结果，并强调它们与其他章节所描述结果之间的关系。

本书编辑 Todd Green 和 Nate McFadden 给了我非常重要的帮助和建议。我要感谢匿名审稿人的真知灼见。书中任何疏漏（包括概念上、技术上或者文体上的）均由我负责。

——Marilyn Wolf

本书旨在为新兴的高性能嵌入式计算领域提供一个参考框架。嵌入式计算机的发展已经远远超越了早期8位微控制器的时代。如今，嵌入式计算机更成为可以运行成千上万行代码的多处理器。它们实时运行并且功耗水平很低。为了更好地设计这类系统，人们对嵌入式软件和硬件特性有关的问题进行了大量持续的研究。现实生活中的飞机、移动电话和数字电视等均依赖于高性能嵌入式系统。我们对如何设计此类系统已有了解，但仍有更多的知识需要掌握。

实时控制实际上是计算机最初的应用之一——第1章中提到的研制于20世纪50年代的MIT Whirlwind计算机就用于武器控制。但微处理器使得嵌入式计算占据了计算机应用领域中较为重要的位置。尽管早在1980年就开始应用复杂的嵌入式系统，但嵌入式计算作为一个学术研究领域直到20世纪90年代才形成。直到今天，许多传统的计算机科学和工程学科在探讨嵌入式计算议题时并未充分了解其他学科研究领域中的相关工作。

嵌入式计算机应用广泛，每年售出数十亿台。大量专业人员设计嵌入式系统——超过50万程序员从事嵌入式软件编程。尽管各种嵌入式系统在细节上有很多不同，但也有一些应用于嵌入式计算领域的通用原理。有些原理早已发现，有些则近期刚被人们知晓。嵌入式计算作为一个研究领域的发展促使嵌入式系统设计从一种技术转变为一门学科，鉴于某些重要的（安全关键的）任务完全交由嵌入式计算机执行，这一转变非常合乎时宜。

对于此领域有理由提出一个问题：它与传统计算机系统的主题有何不同，如客户/服务器系统或者科学计算？我们是否仅仅将相同的理论应用于较小的系统，还是需要做一些新的工作？我相信，尽管嵌入式计算使用了计算机科学和工程中的很多技术，但它也提出了一些独特的挑战。首先，绝大多数嵌入式系统必须执行实时任务。这需要软件和硬件设计者的设计思想有重大转变。其次，嵌入式计算更加重视功率和能量消耗。虽然能耗对于计算机系统的各个方面都十分重要，但与很多通用应用相比，嵌入式应用更加注重功耗、效率。相对于为通用用途设计的计算机，所有这些都导致嵌入式系统需要更为复杂的设计来满足特定的需求。

本书假设读者已经了解了嵌入式硬件和软件的基础知识，如《Computers as Components》中提到的内容。本书在此基础上探讨一系列高级主题。在主题的选择上，我们力图找出嵌入式计算所特有的主题和结论。本书专门介绍了其他学科的背景资料以便于更好地讨论嵌入式系统的问题。

下面是本书简要的介绍。

第1章为本书其他章节提供一些重要的背景资料。本章试图划定嵌入式计算领域中的一组核心主题，着眼于方法学和设计目标。本章综述了各种计算模型，即应用特征的参考框架。本章也对多种依赖于嵌入式计算的重要应用进行了综述，以便于提供全书中将要用到的一些专业术语。

第2章介绍处理器。本章考虑嵌入式系统中使用的不同类型的处理器，讨论调整处理器性能的技术，如电压调整。本章考虑嵌入式CPU中存储层次结构的角色，着眼于优化嵌入式CPU的技术，如代码压缩和总线编码。本章也考虑处理器模拟的技术。

第3章研究程序。编译过程的后端有助于确定代码的质量，这是第一个主题。本章在存储

系统优化上花费大量时间，因为存储行为是性能和功耗的主要决定因素。本章考虑性能分析，包括模拟和最坏情况执行时间分析。本章也考虑计算模型如何反映到编程模型和编程语言中。

第4章讨论多处理系统。本章研究和比较调度算法，包括语言设计和调度机制之间的交互。本章评估操作系统的体系结构和操作系统带来的开销，还考虑验证多处理系统行为的方法。

第5章重点介绍多处理器体系结构。本章考虑紧耦合多处理器和车辆中使用的物理分布式系统，考虑体系结构及其组件：处理器、内存和网络。本章也着眼于多处理器设计的方法学。

第6章着眼于多处理器软件。本章考虑多处理器的调度算法，还研究多处理器中用于动态资源分配的中间件体系结构。

第7章重点讨论软硬件协同设计。本章研究用于描述嵌入式应用和目标平台特性的不同模型，研究很多协同综合算法，并且对这些算法使用的模型和假设进行比较。

希望本书能够涵盖高级嵌入式计算系统的从业者和学生感兴趣的绝大部分主题。有些主题的参考文献很少，嵌入式软件测试就是一个典型例子。我试图为每个问题找到其主要解决方法的代表性论文，但是很遗憾，在很多情况下我无法充分描述一个特定的问题。

本书是关于嵌入式计算的，它略微涉及但没有详细阐述下面的一些相关领域。

应用。嵌入式系统支持多媒体、通信等应用。第1章介绍了一些应用涉及的基本概念，对这些应用领域有一些理解非常重要。关于这些应用领域的深入介绍请查阅其他文献。

VLSI。虽然片上系统是嵌入式系统的重要介质，但不是唯一介质。汽车、飞机和很多其他重要系统都由分布式嵌入式网络控制。

混合系统。混合系统研究连续和离散系统之间的交互。这是一个重要且有趣的领域，很多嵌入式系统可以使用混合系统的技术，但混合系统有专门的文献著述。

软件工程。软件设计是一个内容丰富的领域，它提供了关键的基础，但对于嵌入式计算也遗留了很多尚未解决的问题。

我要感谢帮助我编写本书的所有人：Robert P. Adler (Intel), Brian Butler (Qualcomm), Alain Darte (CNRS), Babak Falsafi (CMU), Ran Ginosar (Gen - Gurion University), John Glossner (Sandbridge), Graham Hellestrand (VaSTSystems), Paolo Ienne (EPFL), Masaharu Imai (Osaka University), Irwin Jacobs (Qualcomm), Axel Jantsch (KTH), Ahmed Jerraya (TIMA), Lizy Kurian John (UT Austin), Christoph Kirsch (University of Salzburg), Haris Lekatsas (NEC), Frank Mueller (NCSU), Pierre Paulin (ST Microelectronics), Laura Pozzi (University of Lugano), Chris Rowen (Tensukuca), Rob Rutenbar (CMU), Deepu Talla (TI), David Whalley (FSU), Jiang Xu (Sandbridge), 以及 Shengqi Yang (Intel)。

我非常感谢编辑 Nat McFadden 以及审校人员对我的支持、指导和鼓励。审校过程有助于明确本书的角色，Nat 始终能给我提供深刻的见解和评论。我还要感谢一直支持我的编辑——Morgan Kaufmann 出版社的 Denise Penrose，他从一开始一直在关注本书的进度。

我还要感谢数字图书馆，特别是 IEEE 和 ACM 的数字图书馆。如果没有它们的帮助本书不可能完成。如果我需要在传统图书馆中查找所用到的所有文献，那我就要身心疲惫地奔走在书架间。有了数字图书馆的帮助，我只需要擦亮双眼。

感谢 Nancy 和 Alec 的耐心，他们是最亲爱的人。

——Wayne Wolf

新泽西普林斯顿

目 录

High-Performance Embedded Computing: Applications in Cyber-Physical Systems and Mobile Computing, Second Edition

出版者的话

译者序

第2版前言

第1版前言

第1章 嵌入式计算 1

1.1 高性能嵌入式计算的现状 1

1.2 信息物理系统和嵌入式计算 3

1.2.1 交通工具控制和操作 3

1.2.2 医疗设备和系统 5

1.2.3 电力 6

1.2.4 无线电和网络 7

1.2.5 多媒体 10

1.3 设计方法 12

1.3.1 为什么使用设计方法 13

1.3.2 设计目标 13

1.3.3 基本设计方法 14

1.3.4 嵌入式设计流程 15

1.3.5 基于标准的设计方法 16

1.3.6 设计验证和确认 18

1.3.7 方法学 18

1.3.8 算法和体系结构联合开发 19

1.4 计算模型 19

1.4.1 为什么学习计算模型 20

1.4.2 图灵机 20

1.4.3 面向流模型 21

1.4.4 状态和控制的表示 23

1.4.5 并行性和通信 24

1.4.6 并行的来源和使用 26

1.5 可靠性、安全性和保密性 27

1.5.1 为什么需要可靠的嵌入式系统 27

1.5.2 可靠系统设计的基础 28

1.5.3 新型攻击和对策 30

1.6 消费类电子产品体系结构 31

1.6.1 蓝牙 31

1.6.2 WiFi 32

1.6.3 网络化消费类电子设备 33

1.6.4 高层次服务 33

1.7 小结和展望 35

本章所学内容 35

进一步阅读 35

习题 35

实验练习 36

第2章 CPU 37

2.1 引言 37

2.2 处理器的比较 37

2.2.1 评价处理器 37

2.2.2 处理器的分类 38

2.2.3 嵌入式处理器与通用处理器 39

2.3 RISC 处理器和数字信号处理器 39

2.3.1 RISC 处理器 40

2.3.2 数字信号处理器 40

2.4 并行执行机制 44

2.4.1 超长指令字处理器 44

2.4.2 超标量处理器 45

2.4.3 SIMD 和向量处理器 45

2.4.4 线程级并行性 46

2.4.5 GPU 46

2.4.6 处理器资源的利用 47

2.5 性能可变 CPU 体系结构 48

2.5.1 动态电压和频率缩放 49

2.5.2 可靠性和错误感知计算 49

2.6 处理器存储层次结构 50

2.6.1 存储器组件模型 50

2.6.2 寄存器文件 54

2.6.3 缓存 54

2.6.4 暂存存储器	56	3.4.3 路径计时	110
2.7 编码和安全	56	3.5 计算与编程模型	114
2.7.1 代码压缩	56	3.5.1 面向中断的语言	115
2.7.2 代码和数据压缩	65	3.5.2 数据流语言	116
2.7.3 低功率总线编码	66	3.5.3 面向控制的语言	120
2.7.4 安全性	69	3.5.4 Java	123
2.8 CPU 模拟	71	3.5.5 计算异构模型	124
2.8.1 基于跟踪文件的分析	73	3.6 小结	126
2.8.2 直接执行	74	本章所学内容	126
2.8.3 微体系结构建模模拟器	74	进一步阅读	126
2.8.4 功耗和热度的模拟与建模	75	习题	127
2.9 自动化 CPU 设计	75	实验练习	128
2.9.1 可配置处理器	76		
2.9.2 指令集综合	82	第 4 章 进程和操作系统	129
2.10 小结	86	4.1 引言	129
本章所学内容	86	4.2 实时进程调度	129
进一步阅读	86	4.2.1 预备知识	129
习题	86	4.2.2 实时调度算法	131
实验练习	87	4.2.3 多关键级调度	136
		4.2.4 动态电压和频率缩放 调度	138
第 3 章 程序	88	4.2.5 性能估计	141
3.1 引言	88	4.3 语言和调度	142
3.2 代码生成和后端编译	88	4.4 操作系统设计	146
3.2.1 指令模型	89	4.4.1 嵌入式操作系统的内存 管理	146
3.2.2 寄存器分配	90	4.4.2 实时操作系统结构	146
3.2.3 指令选择和调度	92	4.4.3 操作系统开销	147
3.2.4 代码放置	93	4.4.4 对调度的支持	148
3.2.5 编程环境	95	4.4.5 进程间通信机制	149
3.3 面向存储的优化	96	4.4.6 功耗管理	149
3.3.1 循环变换	96	4.4.7 嵌入式设备中的文件 系统	150
3.3.2 全局优化	99	4.5 验证	153
3.3.3 缓冲区、数据传输和内存 管理	100	4.6 小结	155
3.3.4 面向缓存和暂存的优化	102	本章所学内容	156
3.3.5 面向主存的优化	104	进一步阅读	156
3.4 程序性能分析	106	习题	156
3.4.1 性能模型	107		
3.4.2 路径分析	108		

实验练习	157	6.2 嵌入式多处理器软件有何 不同	196
第5章 多处理器体系结构	158	6.3 实时多处理器操作系统	197
5.1 引言	158	6.3.1 操作系统的角色	197
5.2 为什么需要嵌入式多处理器	159	6.3.2 多处理器调度	199
5.2.1 嵌入式系统需求	159	6.3.3 动态任务的调度	209
5.2.2 性能和能量	160	6.4 嵌入式多处理器的服务和 中间件	210
5.2.3 专用性和多处理器	161	6.4.1 基于标准的服务	211
5.2.4 灵活性和效率	162	6.4.2 片上系统服务	213
5.3 多处理器设计技术	162	6.4.3 服务质量	216
5.3.1 多处理器设计方法	162	6.5 设计验证	219
5.3.2 多处理器的建模和模拟	163	6.6 小结	220
5.4 多处理器体系结构简介	164	本章所学内容	220
5.5 处理单元	167	进一步阅读	220
5.6 互连网络	167	习题	220
5.6.1 模型	168	实验练习	222
5.6.2 网络拓扑	169		
5.6.3 路由和流控制	171	第7章 系统级设计和软硬件 协同设计	223
5.6.4 片上网络	172	7.1 引言	223
5.7 存储系统	177	7.2 性能估计	223
5.7.1 传统并行存储系统	177	7.2.1 高层综合	223
5.7.2 存储模型	178	7.2.2 加速器估计	226
5.7.3 异构存储系统	178	7.3 软硬件协同综合算法	228
5.7.4 一致性并行存储系统	180	7.3.1 程序表达	229
5.8 物理分布式系统和网络	182	7.3.2 平台表达	230
5.8.1 CAN 总线	183	7.3.3 模板驱动的综合算法	230
5.8.2 时间触发架构	183	7.3.4 通用多处理器的协同 综合	235
5.8.3 FlexRay	184	7.3.5 多目标优化	239
5.8.4 航空网络	189	7.3.6 控制和 I/O 综合	242
5.9 多处理器设计方法和算法	190	7.3.7 存储系统	243
5.10 小结	193	7.3.8 可重构系统的协同综合	244
本章所学内容	194	7.4 电子系统级设计	245
进一步阅读	194	7.5 热度感知设计	247
习题	194	7.6 可靠性	249
实验练习	195	7.7 系统级模拟	250
第6章 多处理器软件	196	7.8 小结	252
6.1 引言	196		

本章所学内容	252	8.5.2 形式化方法	261
进一步阅读	252	8.6 安全性	264
习题	252	8.7 小结	267
实验练习	253	本章所学内容	267
第8章 信息物理系统	254	进一步阅读	268
8.1 引言	254	习题	268
8.2 控制理论和系统	254	实验练习	268
8.3 控制/计算协同设计	255	术语表	269
8.4 网络化控制系统	257	参考文献	287
8.5 设计方法学	260	索引	314
8.5.1 基于模型的设计	260		

嵌入式计算

本章重点

- 嵌入式计算中的基本问题
- 信息物理系统和嵌入式计算
- 嵌入式系统的设计方法和系统建模
- 计算模型
- 可靠性和安全性
- 消费类电子产品

1.1 高性能嵌入式计算的现状

高性能嵌入式计算系统无处不在，如手机、汽车、医生所使用的医疗设备等。它们不仅需要进行大量的计算，还必须满足诸多量化目标：实时性能（而不仅仅是平均性能）、功率/能耗和成本。诸多量化目标使得嵌入式计算系统的设计与通用计算系统有很大的不同，因为后者的用途通常不可预知。

当试图设计满足各种量化目标的计算机系统时，我们很快会发现没有一个系统对所有应用都是最佳的。不同的计算需求导致我们在性能和功率、硬件和软件等方面做出不同的权衡。我们必须设计与实现不同的系统来满足一系列应用的需求。这些解决方案应该具有足够的可编程性以保证设计灵活性和生命力，但又不提供不必要的灵活性以免降低其满足系统需求的能力。

通用计算系统将硬件和软件的设计分开进行，但在嵌入式计算系统中我们能够同时设计硬件和软件。通常，一个问题可以通过硬件方式、软件方式或两者相结合的方式来解决，不同的解决方案有不同的权衡。综合考虑软硬件的设计方法提供了更大的设计空间，允许我们找到更好的解决方案。

如图 1-1 所示，嵌入式系统设计的研究确切地说需要考虑三个方面：**体系结构**（architecture）、**应用**（application）和**方法学**（methodology）。与通用计算机系统的设计相比，嵌入式计算系统的设计人员更依赖于方法学以及与应用有关的基本知识。我们分别考虑这些方面。

因为嵌入式系统设计人员同时与硬件和软件打交道，所以他们必须了解很多体系结构的知识，从广义上来说，即硬件、软件以及两者之间的关系。硬件体系结构问题包括使用软硬件协同设计创建的专用硬件单元、处理器的微体系结构、多处理器或分布式处理器网络等。

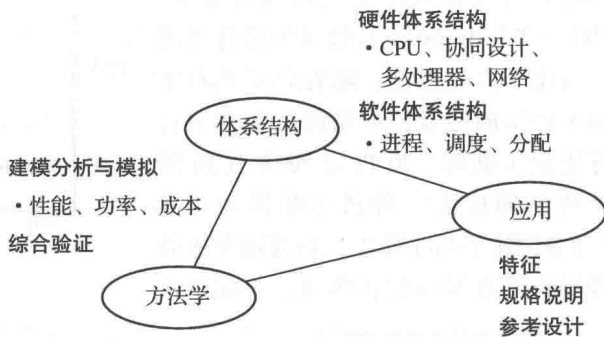


图 1-1 嵌入式系统设计的特征

软件体系结构则决定我们如何利用并行性和非确定性来提高性能与降低成本。

更为有效的设计和使用嵌入式计算系统的关键在于理解你的应用。我们可以充分利用应用的特性来优化设计，实现很多在通用计算机系统中不可能完成的高效优化方案。与此同时，这也意味着我们必须对应用的特性足够了解，才能在充分利用它的同时避免给系统的实现人员制造麻烦。

方法学在嵌入式计算中扮演了一个特别重要的角色。我们不仅需要设计许多不同类型的嵌入式系统，而且希望能构造一个可靠和可预测的设计过程。设计过程本身的花费通常在整个系统的成本中占有很大比重。方法学可以结合工具和人工操作，将我们如何设计系统的知识系统化。方法学有助于我们做出或大或小的设计决策。

通用计算机的设计人员坚持使用以标准基准（如输入、跟踪和模拟）等元素为基础的比较狭义的硬件设计方法。对处理器的修改一般由手工完成，并可能由此产生新的发明。嵌入式计算系统设计人员需要更复杂的方法，因为他们的系统设计涵盖了硬件和软件设计两个方面。嵌入式系统的不同特性——用于通信的片上系统、汽车网络等——也促使设计人员根据他们自己的目标调整设计方法。

设计方法中的各个步骤可以以工具的形式实现。分析和模拟工具已广泛用于评估系统的成本、性能和功耗。综合工具依据系统规格说明来生成优化后的具体系统的实现。工具在嵌入式计算机的设计中显得尤为重要。首先，为特定应用设计系统时，我们可以使用工具来更好地理解应用的特性。其次，设计嵌入式系统时通常时间紧迫，工具可以帮助我们提高工作效率，并由此得到可预测性更好的工具。

嵌入式计算系统的设计越来越依赖于层次化的模型。计算机科学领域多年来一直使用各种模型提供抽象的概念。其中，性能、能耗和功能的抽象尤为重要。因为嵌入式计算系统中的复杂功能建立在各种精细平台之上，所以设计人员必须使用一系列模型才有可能完成系统设计。早期的设计流程需要适当精确的简单模型，后期的设计则需要更复杂和精确的模型。

嵌入式计算系统与很多学科都有关系，其中最为关键的是实时计算和软硬件协同设计。实时系统的研究作为一门学科要早于嵌入式计算出现。实时系统以面向软件的视角，研究如何设计能够完成实时计算的计算机。实时系统中的调度技术也是嵌入式系统设计的核心技术。软硬件协同设计出现在现代嵌入式计算新时代的萌芽期，它以整体的视角来看待运行面向最终期限（deadline-oriented）计算任务的硬件和软件。

图 1-2 展示了嵌入式计算发展的各个重要时期^①。我们可以看到，嵌入式的计算机在计算历史上出现得很早：MIT 的 Whirlwind 作为最早的计算机之一被用于火炮控制。随着计算机科学和工程学的融合，早期研究为实时计算奠定了基础。20 世纪 80 年代到 90 年代期间掀起一股从实时调度技术（如单调速率调度算法）到高端先进处理器的大范围的创新热潮，并催生了



图 1-2 嵌入式计算时间轴

① 本图中的很多数据来源于 Wikipedia。另有一些数据来源于 <http://motofuture.motorola.com> 和 <http://mvista.com>。

从掌上多媒体设备到汽车引擎控制等大量的新产品。当今嵌入式计算的有些技术是专门为解决嵌入式系统中的问题而开发的，有些技术则是由通用计算技术演变而来的。

- 低功率设计最初主要面向硬件，但现在包括软件和硬件技术。
- 编程语言与编译器为嵌入式系统设计人员提供诸如 Java 等工具和高度优化的代码生成器。
- 操作系统不仅提供调度器，还提供文件系统和高性能嵌入式系统中常见的其他工具。
- 网络用于为车辆和许多其他应用创建分布式实时控制系统，以及创建基于互联网的应用。
- 安全性和可靠性成为嵌入式系统设计日益重要的方面。超大规模集成电路（VLSI）组件在尺寸极小的情况下变得不再可靠时，系统对于组件可靠性的需求更加迫切。曾经仅限于通用计算系统中的安全威胁现在也在逼近嵌入式系统。

1.2 信息物理系统和嵌入式计算

了解一些运行在高性能嵌入式计算系统上的应用将有助于我们理解这些系统的设计人员所面临的挑战。信息物理系统（CPS）是包含物理元件和计算元件的一个集成系统。本节将着眼于多个应用中涉及的一些基本概念。其中一些应用为信息物理系统，另一些应用（如多媒体）则强调的是信息和计算。

1.2.1 交通工具控制和操作

交通工具的实时控制是嵌入式计算和信息物理系统的主要应用之一。像汽车和飞机这样的机器需要在其周围部署控制系统。有专门设计的网络用于满足汽车和航空电子设备的实时分布式控制的需要。交通工具的正确操作取决于它的信息元件和物理元件之间正确的相互作用。

为交通工具设计控制系统的基本原因是安全对其至关重要。任何类型的差错——组件失灵、设计缺陷等——都会造成人员伤亡。这些系统不仅需要仔细验证，还必须认真设计以确保某些特定的性能。

车内使用微处理器的最重要应用之一是引擎控制。微处理器引擎控制单元（ECU）的引入是由于提升汽车的燃料效率和控制尾气排放的需求。传统的机械引擎控制机制无法提供适应性、多模式控制以便于获得高燃油效率和低排放量。

引擎控制器的基本工作是控制燃烧的基本元素：空气、燃料和火花。通过向阀门发出指令来控制空气和燃料，通过向打火系统发送脉冲来点燃火花塞。火花塞以一个很高的速率点火。即使火花所需要的计算很简单，服务于火花的指令也需占用大量的 CPU 时间。

控制器有多个传感器。曲轴上的传感器确定引擎转速——曲轴每次到达上止点时即产生一个脉冲。然而，更多的传感器要用于提升引擎的效率。例如，氧气传感器用于确定废气中未燃烧的燃料数量，温度传感器则监测引擎的温度。

微处理器优于机械点火和燃烧系统之处在于使用基于状态的控制。控制系统状态取决于引擎状态：预热、巡航和爬山都是不同的状态。在不同模式下，引擎控制器对于车辆的火花、燃料和空气使用不同的设置，可以通过引擎测量值来进一步调整设置。不同模式下不同控制规则的使用允许引擎避免浪费燃料及减少污染。

如图 1-3 所示，现代的汽车采用了很多种电子设备 [Lee02B]。如今低端汽车通常包含 40 个微处理器，而高端汽车可能使用 100 个微处理器。这些设备通常组织成各种网络。至关重要

的控制系统如发动机和刹车控制系统可能位于一个网络中，而娱乐设备等非关键功能则可能在另一个网络中。

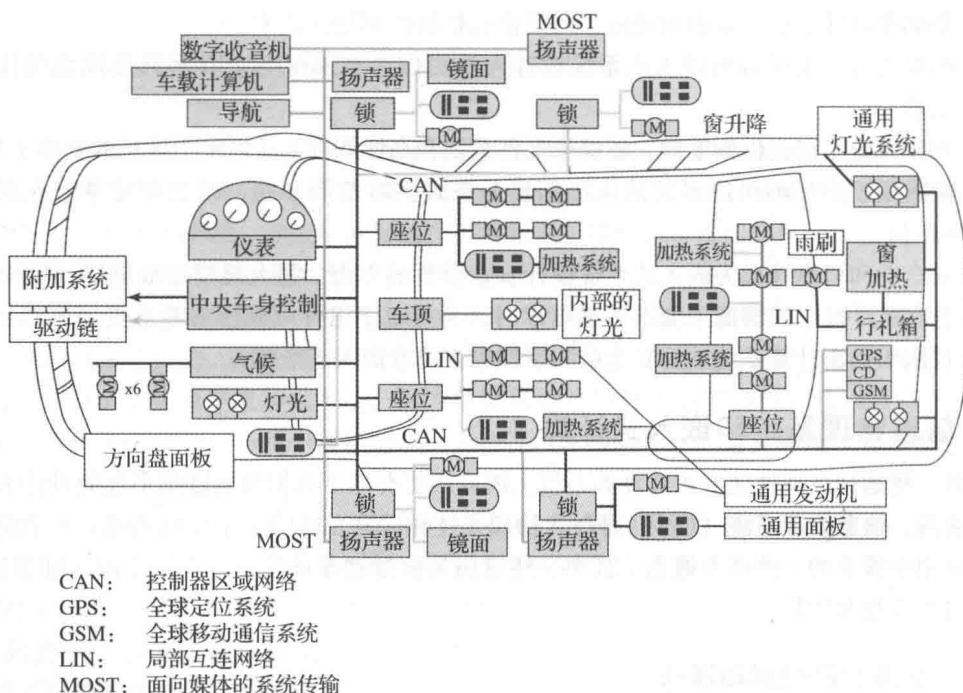


图 1-3 现代汽车中的电子设备，图片摘自 Lee [Lee02B]

在数字电路技术到来之前，汽车通常采用点到点、成捆线缆的**线束**（harness）方式布线。使用共享网络的方式连接设备大大降低了线缆的重量——15 千克或者更多 [Lee02]。网络需要更复杂的控制设备，包括网络访问硬件和软件，但其带来的开销相对较小并且根据摩尔定律时间还在不断缩短。

但为何不在实时控制中使用通用的网络？我们有理由在网络协议栈的各个抽象层级上构建专门的汽车网络。一个原因是与电子信号有关——汽车网络需要在非常苛刻的环境下保持可靠的信号传输。汽车引擎的点火系统会产生大量可导致网络失灵的电磁干扰。同时，汽车还必须在较宽的温度范围和非常潮湿的环境下运行。

最重要的是，实时控制需要网络提供有保证的行为（guaranteed behavior）。很多通信网络不能满足硬实时的需求。通信系统比控制系统更能容忍延迟。当网络传输产生数百毫秒甚至数秒的延迟时，数据或语音通信可能仍然有用，但在实时控制系统中的长延时容易引起灾难性的振荡。汽车网络还必须运行在有限功率预算之下，这在通信网络中不常见。

航空电子系统与汽车电子系统同时发展并逐渐互相融合。航空电子设备必须经过政府机构认证才可用于飞行器（美国联邦航空管理局，FAA），这意味着飞行器的设备通常是专门为其设计的。事实上，经过认证的航空系统更容易在类似于飞行器操纵面（如副翼、方向舵、升降舵）等关键操作上使用电子设备。飞机驾驶舱也高度自动化。一些商用飞机已经为乘客提供互联网接入功能，我们期望此类服务在下个十年能在汽车中普及。

控制系统一直以来依赖机械或者水力部件来实现反馈和反作用。微处理器不仅允许我们用硬件和软件来感应与驱动，还可以实现各种控制规则。一般来说，控制器在物理位置上可能不在被控制设备的附近：控制器可能操作若干个不同的设备，或者它可能有意避开危险的操作环

境。关键功能的电子控制首先在战斗机中运用，这就是**线控飞行**（fly-by-wire）。通过网络完成的控制操作称作**线传控制**（X-by-wire），这里的X可能是刹车、方向盘等。

强大的嵌入式设备（如电视系统、导航系统、因特网访问等）正引入到汽车内部。这些设备不执行实时控制，但它们占用大量的带宽并且需要为数据流提供实时服务。因为汽车内传输的数据量会不断增加，所以汽车网络的设计必须有长远考虑，并能处理比现在更重的负载。

通常，我们可以将交通工具中的网络应用根据以下方法分类。

- **操作员与乘客**（operator versus passenger）——这是交通工具网络中最基本的特性。乘客可能因为各种目的使用网络：娱乐、信息等。但乘客的网络不能干扰交通工具驾驶或者飞行所需的基本控制功能。
- **控制与仪表**（control versus instrumentation）——交通工具的运转依赖很多设备。基本的控制功能——汽车中的转向、刹车、油门等或者飞机中的操纵面和油门——必须反应迅速且完全可靠。操作员用到的其他功能可能不那么重要。虽然飞机中一些仪表对监控飞行中的气象条件是非常重要的，但飞行员通常只需识别一小部分仪表信息即可控制飞机。驾驶汽车的时候，则更少关注仪表。虽然仪表非常重要，但我们要将它从基本控制中分离以保护控制系统的运转。

1.2.2 医疗设备和系统

医疗系统可谓是复杂的系统的系统的典型。单个器械扮演了复杂的作用，但医疗工作人员需要这些器械共享信息。Hatcliff 等 [Hat12] 描述了一个应用于网络化医疗设备的**医疗应用平台**（Medical Application Platform, MAP）。MAP 具有如下多种多样的功能。

- 应用可以在设备间或向电子医疗记录系统传输数据。支持的设备包括个人电脑、智能手机和传统的医疗设备。
- 应用程序可以提供成熟的算法来决定何时发出报警音，如使用血氧测定法。
- MAP 支持应用程序通过集成来自不同源的信息来做出临床决策。
- 该平台支持基于所测数据采取行动的闭环控制。
- MAP 支持工作流工具来指导医疗人员执行标准流程。
- 平台需要提供用于仪器内以及仪器间的安全互锁的工具。

7

采用医疗应用平台面临几个问题。

- 当前互操作性标准不支持实时控制、安全性、保密性和连通性。
- 当前系统不鼓励使用多个供应商提供的组件以即插即用的方式部署系统。
- 在美国，医疗设备必须获得食品药品监督管理局（FDA）认可。此种监管流程不适合开放平台架构。

图 1-4 展示了集成临床环境（ICE）的架构 [MDP12]。管理器提供虚拟机环境用于执行应用。在虚拟机上建立应用的做法提供额外的隔离措施来保障安全性。网络控制器追踪设备间的通信和

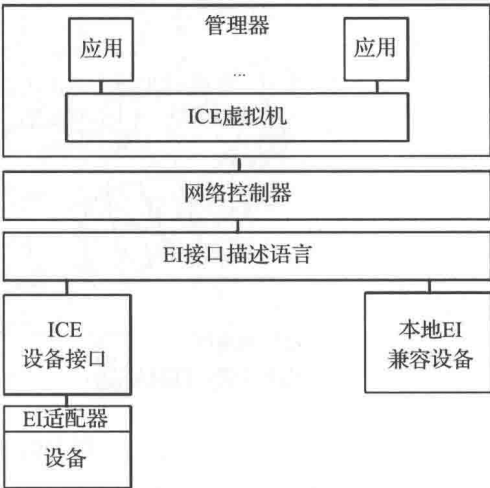


图 1-4 ICE 医疗应用平台的架构