



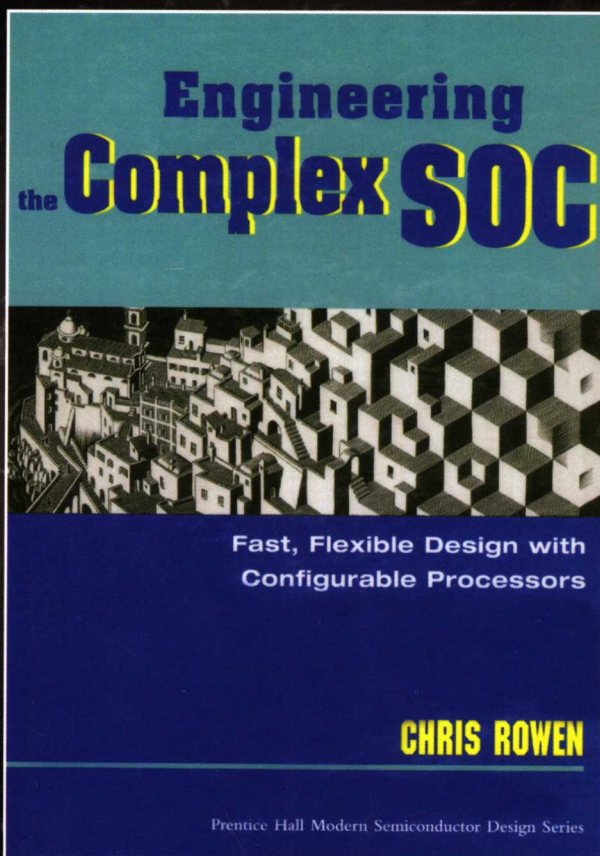
计 算 机 科 学 丛 书

复杂SOC设计

(美) Chris Rowen 著

吴武臣 侯立刚
北京工业大学集成电路实验室

译 陈弘毅 审校



Engineering the Complex SOC
Fast, Flexible Design with Configurable Processors



机械工业出版社
China Machine Press

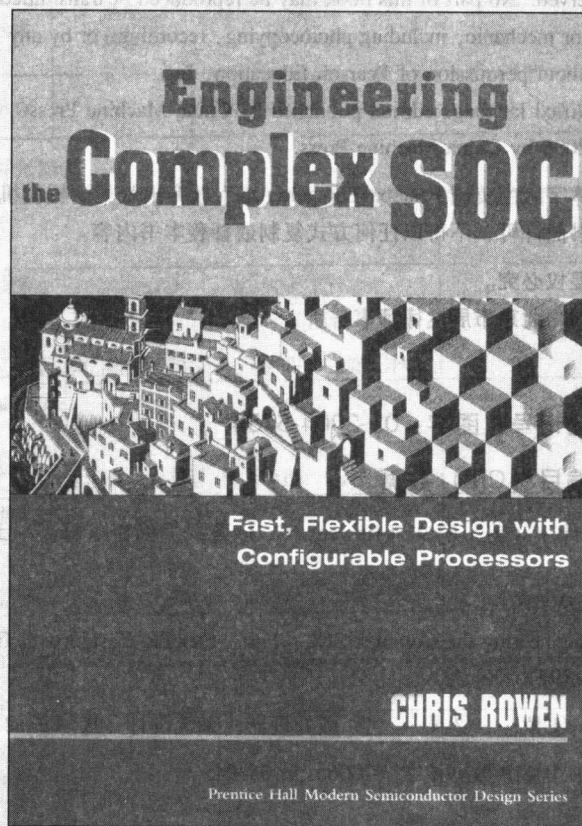
计 算 机 科 学 丛 书

复杂SOC设计

(美) Chris Rowen 著

吴武臣 侯立刚
北京工业大学集成电路实验室

译 陈弘毅 审校



Engineering the Complex SOC
Fast, Flexible Design with Configurable Processors



机械工业出版社
China Machine Press

本书综合介绍可扩展的处理器架构、用于指令集扩展的工具以及用于嵌入式系统的多处理器 SOC 架构。全书共分三部分, 第一部分对众多 SOC 设计问题及其解决方案进行了高层次的介绍。第二部分对可扩展的处理器、传统处理器以及硬连线的逻辑电路之间的比较进行了详细的讨论。最后一部分则给出一系列详细的例子, 来进一步说明新的 SOC 设计方法的可用性。全书使用了来自 Tensilica 公司的 Xtensa 架构和 Tensilica 指令扩展语言, 精确描述与这个新方法相关的实践问题和机遇。

本书主要面向从事复杂 SOC 设计的架构设计师、电路设计师和程序设计师, 也可供高等院校相关专业师生参考。

Authorized translation from the English language edition entitled *Engineering the Complex SOC: Fast, Flexible Design with Configurable Processors* by Chris Rowen, published by Pearson Education, Inc, publishing as Prentice Hall PTR (ISBN0-13-145537-0), Copyright © 2004 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2006 by China Machine Press.

本书中文简体字版由美国 Pearson Education 培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2004-5754

图书在版编目 (CIP) 数据

复杂 SOC 设计 / (美) 罗文 (Rowen, C.) 著; 吴武臣等译. - 北京: 机械工业出版社, 2006. 7

(计算机科学丛书)

书名原文: *Engineering the Complex SOC: Fast, Flexible Design with Configurable Processors*
ISBN 7-111-19215-X

I. 复… II. ①罗… ②吴… III. 微处理器 - 系统设计 IV. TP332

中国版本图书馆 CIP 数据核字 (2006) 第 053945 号

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: 朱起飞

北京慧美印刷有限公司印刷·新华书店北京发行所发行

2006 年 8 月第 1 版第 1 次印刷

184mm × 260mm · 21.25 印张

定价: 42.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

本社购书热线: (010) 68326294

出版者的话

文艺复兴以降，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；也正是这样的传统，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，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann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Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及度藏。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襄助，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，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

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,但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。教材的出版只是我们的后续服务的起点。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,我们的联系方法如下:

电子邮件: hzjsj@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

译者序

目前, 微电子技术的发展变化异常迅速, 是任何技术和产业所不能比拟的。进入 21 世纪以后, SOC 成为微电子技术领域与微电子产业界关注的焦点。人们普遍认为, SOC 将是集成电路的发展方向, 它的发展将进一步扩大信息产业的应用领域, 给信息产业带来巨大的发展空间与丰厚的利润。也许有人清楚地记得, 20 世纪 80 年代, ASIC 的兴起给微电子产业带来了巨大影响: 它促进了集成电路产业向“设计、制造和封装”三业并举的方向发展, 推动了 EDA 产业的兴起, 促进建立了代工机制及专业的代工厂和设计公司。SOC 的影响不论从深度和广度都将超过 ASIC, 因为除了影响集成电路产业外, 它还将波及整机产业, 甚至整个信息产业。而这一产业组成的变革也将反过来推动产业本身技术的发展。

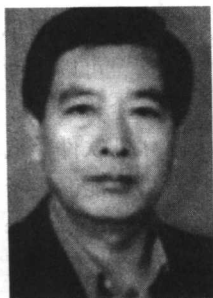
对于 SOC 这个新兴的产业, 目前还很少有合适的教科书和参考资料。这本全面的、综合的教材将帮助读者在这个蓬勃发展的事业中, 成为一个成功的参与者。本书以崭新的视角, 详细地讨论了复杂 SOC 的设计, 内容包括从市场推动力到技术挑战; 从专用逻辑电路的集成, 到通过对处理器的定制化来使性价比最大化的技术方法。本书并不是用同等笔墨强调所有的设计方法, 而是专注于讨论这种新的 SOC 设计方法的优点, 强调面临的机遇, 并讨论从以门为中心的设计到以处理器为中心的 SOC 设计方法的转变问题。书中使用来自 Tensilica 公司的 Xtensa 架构和 Tensilica 指令扩展语言, 以大量详实的范例, 深入浅出地阐述了这种全新的设计理念。

本书适合从事复杂 SOC 设计的架构设计师、电路设计师和程序设计师研读, 并可供 SOC 设计公司的管理者和对 SOC 设计感兴趣的高等院校的高年级本科生和研究生参考。对这些“非专家”读者来说, 理解这些设计过程和硬件及软件的基本面, 对于商业决策或技术方向选择也非常有参考价值。

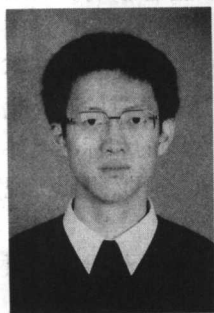
本书是在北京工业大学 VLSI 与系统研究室主任、博士生导师吴武臣教授的组织下翻译的。吴武臣老师具体指导并亲自参与了全书的翻译工作, 还审阅了本书的全文。北京工业大学的侯立刚博士和清华大学微电子学研究所的陈刚博士在全书的翻译过程中做了大量工作。参与翻译工作的还有: 宋云扬、罗仁贵、冯哲、邓璐、李京波、李莱、刘成、张乃燃、谷威、谢通、吴晓鹏、王继祥、周屹、刘彤芳、王旭、王洪波等。清华大学微电子所的博士生导师陈弘毅教授于百忙之中详细地审校翻译初稿, 并对翻译工作提出了很多宝贵的指导意见。

由于本书内容涉及大量的新技术和新概念, 加上译者水平有限, 时间又非常紧迫, 因此错漏之处在所难免, 恳请读者批评指正。

译者简介

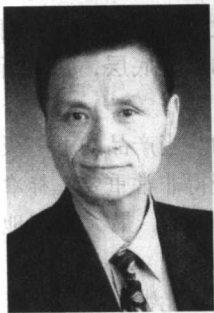


吴武臣 教授，博士生导师，北京工业大学 VLSI 与系统研究室主任。1981 年于北京工业大学电子工程系获硕士学位，后留校从事教学科研工作。1988~1989 年在瑞士苏黎世工大作访问学者，1992~1995 年任苏黎世工大客座教授，多次应邀在瑞士、德、法、日本和意大利等国作研究报告。自 2000 年来，从事集成电路设计与系统集成研究工作并创建 VLSI 与系统研究室。先后承担国家自然科学基金、北京市自然科学基金以及北京市教委科研项目多项，并获电子部科技进步一等奖一项及其他奖项。发表论文 60 多篇，专著、译著各一部。



侯立刚 博士，2006 年毕业于北京工业大学微电子学与固体电子学专业，获博士学位。研究方向为大规模集成电路与系统集成。主持多个芯片的设计流片，有丰富的数字、混合信号芯片设计经验。目前主要从事芯片设计、低功耗设计方法和 EDA 工具等方面的研究工作。

审校者简介



陈弘毅 教授，博士生导师，清华大学学术委员会委员、信息学院学术委员会副主任、微电子学研究所学术委员会主任，IEEE 高级会员、中国电子学会高级会员。多年来从事集成电路（IC）设计研究，研究方向包括：专用集成电路（ASIC）设计方法；库开发技术；系统芯片集成；算法的硬件架构与实现；超大规模集成电路-数字信号处理等。参加并主持国家科技攻关、863 重点项目和国家自然科学基金等多项计划，获得国家科技进步二等奖一项、北京市科技进步二等奖两项；获准持有美国发明专利 13 项；发表学术论文约 200 篇；参与出版专著两部：《集成电路工业全书》和《ULSI 器件、电路与系统》，译著一部：《VLSI-DSP 系统——设计与实现》（机械工业出版社，2004）。

序 言 一

这是一本很重要的、开卷有益的好书。说它重要，是因为它讨论了一种机遇，这种机遇迟早会出现在每一个行业；说它开卷有益，是因为它提供了一种清晰的、循序渐进的方法，可以帮助微电子业界的工程师和管理者们抓住这个机遇来创造财富，实现利润的增长。抓住这个机遇的公司，将能够转变业内竞争的模式。有很多其他行业也开始出现这种机遇——从计算机软件、汽车、通信设备，到哈佛大学商学院开设的管理类专业。我希望这本深入浅出的指导书也能够用于帮助这些业界的战略家和工程师们，这种业界的变化的现象称为竞争基础的转变（a change in the basis of competition）——这个变化是某种形式的产品和服务的改进，使得顾客愿意为此付出更高的价格。

这是一个很自然的和可预测的过程，在这个过程中发生的变化对业界产生着影响。Chris Rowen 是一位很有天赋的战略家、工程师和企业家，几年以来，他一直和我一起努力工作，试图能够更好地理解这一过程。在这个过程的开始阶段，某个公司开发出专有的产品，尽管性能还不够好，但是和其他竞争对手相比，相对更符合顾客的需要。最为成功的公司，通过专有的和优化的架构来实现这个目标，因为在这个阶段，这些产品的功能性和可靠性比使用公开的和标准组件的产品要好。为了能够提供具有专有的、架构上互相关联的产品，最成功的公司之间必须要实现纵向一体化。

然而，公司继续努力保持其对直接竞争对手的优势，最终将会超越市场上要求不是很高的顾客层对功能性和可靠性的需求。这就促进了在这些领域内竞争基础的改变。顾客将不再为更好、更快和更可靠的产品付出更高的价格，因为他们并不需要这些方面的产品性能改善。这时，不足之处就变成了速度和方便性。顾客开始期望有新的产品能够积极满足他们定制构造的需要，并且能够尽可能快速和方便地设计出来并交付使用。因此，创新的方向是，要争取使产品的改进有利于有吸引力的产品价格并改变市场份额。为了在这方面进行竞争，即：快速、灵活和对市场反应灵敏，产品的主要构造必须要向标准化组件方向发展，其元件和子系统的接口必须要符合业界的标准。当实现了这个目标之后，领先公司之间的纵向一体化就变得没有什么优势。由于关键的接口标准已经完整、清晰地给出，元件和子系统的供应商可以开始独立地开发、制造和销售他们的产品，能方便地和合作伙伴及顾客打交道。这种情况在最初出现功能过剩的市场底层，并逐渐向上移动，最终不可避免地会影响到更高层的市场。

当产品或子系统的架构是由标准组件构成时，那么它是符合适应性要求的。当产品的下一级功能性和可靠性还未能完全满足顾客要求的产品组合在一起时，这种适应性变得非常重要。标准组件化的适应性（modular conformability）使得顾客能够对其购买的产品进行定制，最大程度地获得自己所需要的功能，避免不需要的功能。

微处理器行业目前正在精心地进行这种过渡。微处理器，以及它所实现的功能数，从历史上看还不够令人满意，因此，它的架构一直是专有化的和经过专门优化的。然而，有足够的证据表明，对市场的主流需求来说，竞争的基础正在发生变化。微处理器对绝大多数的用户来说已经足够快速了。为了追赶摩尔定律（Moore's Law）的如期实现，芯片加工厂已经将晶体管的特征尺寸降低到非常小的地步，以至于对市场的大部分需求层次，电路设计者已经很难应付如此多的晶

体管，他们没有能力使用摩尔定律给他们带来的那么多的晶体管。因此，特别是在嵌入式、移动和无线应用领域，顾客定制配置的处理器占据主流地位。它们的标准组件可配置性帮助设计者可以对嵌入处理器的产品系统进行性能优化。

伴随着这个变化，微处理器行业的发展步伐正在加速。产品的设计周期在相互依赖架构时代需以年计，现在已经突降至以月计，而在未来，将以周计。在组成电路的模块之间清晰的标准组件接口，即可复用的 IP 库，最终将使得并不精通处理器设计的工程师也能够设计他们自己的电路。最终，软件工程师将能够设计全定制的处理器的性能进行优化。

在电路设计者和芯片加工厂之间的更清晰的接口标准正在规定之中，它将打破纵向一体化的行业结构，并实现非纵向一体化。这种结构最初在低端市场出现，现在已经占据市场的主流。根据标准组件进行的“订单设计”（designed-to-order）的处理器出现，已经成为微处理器行业发展的一个重要里程碑，但是标准组件化处理器（modular processor）有着更加广泛的涵义。正如个人计算机的出现使得很多工作人员可以第一次使用计算机来组织他们的工作任务，可配置的处理器出现也将改变很多芯片设计者和芯片用户的工作方式。这种新的基于处理器的方法使得设计师和用户可以对处理器进行定制和编程，以便应对那些对成本或能量效率非常敏感以至于传统处理器很难完成的任务。如果高度专业化的微处理器架构和开发团队进行通力协作，就可以使普通的软件或硬件开发工程师能够创造新的计算引擎。这些新的处理器模块可能被大量地使用，在一个芯片内包含几十个甚至上百个处理器，总的可配置处理器数远远超过传统的微处理器。因此，未来将会和过去有很大的差异。

Chris Rowen 在本书中描述的设计原理和技术对希望充分利用这些变化创造新的财富增长的公司，是极为有用的。我非常感谢他为半导体业界带来的这份礼物。

Clayton M. Christensen

Robert 和 Jane Cizik 工商管理学教授

哈佛大学商学院

2004 年 3 月

序 言 二

在过去的 30 多年中，摩尔定律一直是计算机和电子工业发展的推动力，不断地促进变化，驱动创新，使得我们可以将更大的系统集成到一个芯片上。从大约 20 年前开始，摩尔定律就驱动微处理器迅速发展，成为计算机领域内的主导。当前，摩尔定律也是片上系统（system-on-chip, SOC）发展的动力，它使用大幅度增长的晶体管密度来将更多的系统元件集成到一个芯片上，因此可以降低成本、功耗和物理尺寸。

作者 Rowen 围绕着六个基本问题，在这本书中详细地讨论了复杂 SOC 的设计，其内容包括从市场驱动力（例如紧迫的上市时间和在芯片产品过时之前有限的产量）到技术挑战（在坚持一个积极进取的日程表下要达到满意的性能和成本）。这六个挑战使我们专注于 SOC 设计过程的某些特殊方面，从专用逻辑电路的集成，到通过对处理器的定制化来使性价比最大化。

正如 Rowen 所清楚地指出，处理器及其设计是任何复杂 SOC 的核心内容。总之，最为软件化的解决方案可能会减少任务实施时间，降低风险；困难在于这样的解决方案通常不能达到满意的性能或效率。对绝大多数的应用来说，需要有执行专用代码的处理器和专用硬件的组合。在第 4 章，作者讨论了定制的硬件和嵌入式处理器核之间接口的关键问题，他指出：将定制的通信和互连硬件和软件结合起来以处理决策问题和不同寻常的任务，使得设计者可以取得期望的设计努力、风险、性能和成本之间的平衡。

第 5 章和第 6 章组成了本书的核心部分，它是基于 Tensilica 公司几年来在全定制处理器基础上设计 SOC 的经验。尽管设计全定制处理器的潜在优势是很清楚的：成本更低且性能更好，但是，需要的 CAD 工具、可定制的功能块和软件，这些在以前都不存在。现在，这些条件都开始具备，Rowen 讨论了如何使用这些最尖端的工具进行软件和硬件设计，用于复杂 SOC 的实现。

第 7 章和第 8 章讨论了在这样的系统中获得高性能所面临的挑战。对微处理器来说，性能的提高主要来自于并行（parallelism）处理的开发，它是这两章的核心课题。第 7 章讨论了如何在一个处理器的一个指令流内通过使用流水线技术来实现更高的性能。第 8 章则对未来进行了展望。在未来，将越来越多地使用多处理器，并根据应用的需求对这些处理器进行配置和连接。多处理器的使用在很大程度上代表了高性能计算机架构的发展方向，而不仅局限于嵌入式应用。我非常高兴地看到，SOC 在未来的发展蓝图中正在扮演着越来越重要的角色。

对于新的和更有挑战性的应用，从通信到信息设备，以及我们现在还不敢想象的新奇应用，SOC 设计给信息处理带来了新的机遇。本书内容全面、综合性强，将帮助你在这个蓬勃发展的事业中，成为一名成功的参与者。

John Hennessy
斯坦福大学 校长
2004 年 3 月

前 言

这本书是为从事复杂 SOC 设计的架构设计师、电路设计师和程序设计师编写的。那些对 SOC 设计和平台进行重大投资的公司的经理们也将发现，理解这些设计过程和硬件及软件的基本变化是非常重要的。这些变化可能将直接或间接地影响投资策略、核心竞争力，以及组织结构。

本书描述了驱动 SOC 设计过程变化的主要动力，并介绍了使用可扩展的处理器（extensible processor）作为基本设计结构的 SOC 设计概念。本书讲述了可扩展的处理器架构，用于指令集扩展的工具，以及用于嵌入式系统的多处理器 SOC 架构。在整本书中，使用了来自 Tensilica 公司的 Xtensa 架构和 Tensilica 指令扩展（Tensilica Instruction Extension, TIE）语言，以给出与这个新方法相关的实践问题和机遇的最新精确描述。你将能够读到大量关于 Xtensa 和 TIE 的信息以便理解设计方法，尽管这本书并不是 Xtensa 和 TIE 的全面的产品文档。

本书并不是等地强调所有的设计方法。事实上，它专注于讨论这种新的 SOC 设计方法的优点，强调面临的机遇，以及讨论从以门为中心的设计到以处理器为中心的 SOC 设计方法的转变问题。

本书的第一部分对很多 SOC 设计问题及其解决方案进行了高层次的介绍。中间部分对可扩展的处理器和传统处理器以及硬连线的逻辑电路之间的比较进行了更为详细的讨论。它同时讨论了处理器的可扩展性的基本机理是如何应对先进 SOC 架构的计算和通信需求的。后一部分则给出了一系列的详细例子，来进一步说明新的 SOC 设计方法的可用性。

第 1 章介绍了 SOC 设计的一些基本问题，以及为什么要对 SOC 开发过程中硬件和软件结构和设计方法进行革新。

第 2 章对当前的 SOC 硬件结构、软件组织，以及芯片开发流程进行了评述。该章将讲述当前 SOC 设计方法中存在的 6 个基本缺陷，并解释为什么需要新的结构和设计过程。

第 3 章介绍了在芯片上所有控制和数据处理功能中使用可扩展处理器的新的 SOC 设计方法。它简单地讨论了这种方法是如何应对前述的 6 个重大问题的。

第 4 章使用自顶向下的方法研究了以处理器为中心的 SOC 架构，着眼于复杂系统架构中的整体数据流。在这一章中，描述了如何将复杂的功能拆分成功能块，这些功能块通常都可以用专用处理器（application-specific processor）的形式来实现。关键的问题包括迟滞、模块的吞吐率、并行功能之间的协调性编程模型、硬件互连选择，以及在整个芯片设计中的复杂性管理等。

第 5 章和第 6 章则深入讨论了各种各样不同任务的设计。第 5 章侧重于软件开发人员的设计任务，特别是将一个原计划在通用处理器上运行后来改在专用处理器上运行的设计过程。这一章描述了专用处理器是如何满足传统的嵌入式处理器核的使用模型，同时又加入了能够大幅度提高复杂任务的性能和效率的简单机制。这一章还包括了对 Xtensa 架构原理的简单介绍，包括灵活长度指令扩展（Flexible Length Instruction Extension, FLIX）和全自动指令集（fully automated instruction-set）的生成。

第 6 章侧重于硬件开发人员的设计任务，特别是将硬件功能落实到专用处理器，并具有可比的性能和完全的可编程性。这一章确立了硬件流水线和处理器流水线之间的基本对应关系，并推

荐了将传统的硬件功能（包括高带宽、低迟滞的功能）高效地映射到专用处理器的技术。

第7章研究了一系列更加先进的SOC设计课题和问题，包括实现复杂状态机的技术、任务间通信和同步的选择方案、处理器和其他硬件模块之间的接口、功率优化，以及TIE语言的细节等。

第8章，也就是最后一章，展望了远期未来的SOC设计，研究了设计方法和半导体技术的基本趋势。描述了10到15年后，在设计、应用和电子工业结构等方面的定性和定量的变化。

本书在讨论中使用了一系列与SOC设计相关的术语。SOC设计方法是模块、设计生成器、架构指导、工具、仿真方法和分析技术等要素的有机结合，构成了一个能够用于芯片设计和相应软件开发的和谐环境。一般情况下，本书将书中推荐的方法称为先进SOC设计或以处理器为中心的SOC设计方法（processor-centric SOC design methodology）。偶尔也会使用MPSOC设计方法（MPSOC methodology）这个术语，它指的是多处理器片上系统设计方法（multiple-processor system-on-chip design methodology），用以强调处理器的重要作用，通常作为基本模块在灵活SOC设计中大量地使用。最终的发展蓝图是：可配置的处理器变得非常普遍、非常自动化，并且无处不在，以至于我们可以称之为“处理器海”（sea of processors）。在10年之内，处理器可能会成为新的逻辑门，每个芯片上将包含有成百上千个处理器，并且可以对这些处理器进行专用的配置，就像我们当前做的逻辑综合一样成为日常的设计工作。

本书涉及一系列的硬件、软件和系统设计问题，但是并不期望能够覆盖这些领域内的所有内容。为保证内容的连贯性，我们没有在正文中插入很多注释或技术参考文献，但是在每章结尾处加入一节内容用于指导读者的进一步阅读。这一节重点介绍了与该章内容相关的技术参考文献，并且列出了一些参考书目以加强读者对这方面内容的理解。

本书使用Tensilica公司的Xtensa处理器架构和工具阐述重要的设计思想。然而，本书并不试图全面介绍Tensilica公司的产品。如果需要更多的关于Tensilica的细节，请参见网址<http://www.tensilica.com>。本书也介绍了其他的自动处理器生成的方法，特别是在第3章。

致谢

本书是我们一年多努力的成果。很多人以多种方式参与了本书的工作，因此，向这些帮助过本书的编纂的人致谢是一件非常繁重的工作。

然而，从哪里开始致谢，却是很明显的。Steve Leibson，是本书的编辑和我在Tensilica公司的同事，他对本书的定义和指导，起到了核心的作用。他对读者的需求有着清晰的认识，并且他致力于使本书的措辞和概念都更加准确，极大地提高了本书的质量。他对本书草稿的艰苦审校和对本书内容的生动讨论，不仅对提高书稿的水平很有帮助，而且还在很大程度上改变了我对开发新的片上系统平台的芯片设计者的长期需求的看法。如果没有Steve的大量努力和长期坚持，本书是不可能问世的。

Paula Jones作为本书的项目经理，和出版商一起工作，管理本书内容的审校，编辑图片和进行版面设计，因此，需要对她进行特殊感谢。当本书的进度看起来慢得令人痛苦时，她无尽的乐观态度帮助本书顺利度过这些困难的时刻。

有几个同事的工作特别值得提及。在第5章的开放安全套接层（Open Secure Socket Layer）实例研究，主要是由Jay McCauley做的工作。第6章的ATM分段和重组的实例，是基于Jerry Reddington的工作。Himanshu Sanghavi在TIE实例方面花费了很多时间，以确保不仅句法正确，而且更重要的是概念上要正确。Dror Maydan在本书的系统和软件问题上的讨论起到了非常重要

的作用，并且在实例和讨论方面进行了重要的实质性改进。

很多 Tensilica 公司的其他的同事都花费了不少时间来阅读和审校本书的多个版本的草稿，或对内容提出了很重要的建议。我非常感谢 Beatrice Fu、Eileen Peters Long、Bill Huffman、John Wei、David Goodwin、Eliot Gerstner、Steve Roddy、Gulbin Ezer、Jerry Reddington、David Jacobowitz、Robert Kennedy、Chris Songer、Marc Gauthier、Jean Fletcher、Morteza Shafiei、George Wall、Larry Pryzwara、Leo Petropoulos、Tomo Tohara、Nenad Nedeljkovic 和 Akilesh Parameswar 等人的辛勤劳动。

我对三位特别重要的良师益友致以最真诚的谢意，他们是：斯坦福大学的 John Hennessy、伯克利的 Richard Newton 和哈佛大学的 Clayton Christensen。这三位专家扮演了双重的角色——鼓励本书工作的进行和挖掘书中的概念，并提供了重要的评述，以帮助在更大的商业和技术范畴内进行可配置处理器和多处理器片上系统的设计。

最后，本书反映了在 Tensilica 处理器生成器及相关的用于复杂 SOC 设计的工具方面投入的巨大努力。我感谢 Tensilica 的同事们不倦的设计工作，以及他们对本书的支持，使本书得以及时付梓。

Chris Rowen

2004 年 3 月

目 录

出版者的话	
专家指导委员会	
译者序	
译者简介	
审校者简介	
序言一	
序言二	
前言	
第 1 章 为什么需要新的 SOC 设计方法学	1
1.1 百万门 SOC 的时代	1
1.1.1 摩尔定律意味着机遇和危机	2
1.1.2 发展障碍 1: 设计了错误的芯片	3
1.1.3 发展障碍 2: 用错误的方法设计芯片	3
1.2 SOC 设计的基本趋势	4
1.2.1 为每个系统设计一个新的 SOC 是糟糕的想法	5
1.2.2 SOC 设计改革: 更低的设计成本和更大的设计灵活性	5
1.2.3 并发性	6
1.2.4 可编程性	7
1.2.5 可编程性和效率	7
1.2.6 SOC 设计成功的关键: 特定领域的灵活性	10
1.3 当前的 SOC 设计方法存在什么问题	11
1.3.1 传统的处理器存在什么问题	12
1.3.2 传统的 SOC 设计方法存在什么问题	13
1.4 概述: 改进的 SOC 设计方法学	15
1.4.1 SOC 设计流程	15
1.4.2 可配置的处理器作为组成模块	16
1.4.3 一个普通的例子	17
1.4.4 专用处理器配置的结果	17
1.4.5 处理器作为 SOC 的组成模块	19
1.4.6 解决系统设计问题	19
1.4.7 改进的 SOC 设计方法的涵义	22
1.4.8 向基于处理器的 SOC 设计的转变	24
1.5 进一步的阅读资料	24
第 2 章 当前的 SOC 设计	26
2.1 硬件系统结构	26
2.1.1 当前 RTL 的使用情况如何	27
2.1.2 控制、数据通路和存储器	28
2.1.3 硬件趋势	29
2.2 软件结构	29
2.3 当前的 SOC 设计流程	32
2.4 半导体经济学的影响	35
2.5 SOC 设计的 6 个主要问题	36
2.5.1 不断变化的市场需求	36
2.5.2 不足的产品产量和寿命	37
2.5.3 半导体供应链的不灵活性	37
2.5.4 不足的性能、效率和成本	38
2.5.5 设计和验证中的风险、成本和延迟	38
2.5.6 硬件和软件设计团队之间协作不足	39
2.5.7 6 个问题的解决	40
2.6 进一步的阅读资料	40
第 3 章 SOC 设计的新视角	41
3.1 以处理器为中心的 SOC 架构基础	41
3.2 为传统的软件任务加速处理器	45
3.2.1 通用处理器的发展过程	45
3.2.2 阐述可配置性和可扩展性	46
3.2.3 处理器可扩展性	46
3.2.4 设计者定义的指令集	47
3.2.5 存储器系统和可配置性	48

3.2.6 可配置处理器的起源	49	主要决定	85
3.3 例子: 用 EEMBC 基准测试		4.2.1 出发点: 基本接口和计算	86
Tensilica Xtensa 处理器	50	4.2.2 并行化任务	88
3.3.1 EEMBC 消费类基准	51	4.2.3 将任务分配到处理器上	89
3.3.2 电信	52	4.2.4 选择合适的通信结构	93
3.3.3 EEMBC 网络基准	53	4.3 通信设计 = 软件模式 + 硬件互连 ...	96
3.3.4 作为 RTL 替代品的处理器	54	4.3.1 软件通信模式	97
3.4 多处理器系统设计	56	4.3.2 消息传递	97
3.4.1 可用的并发性	56	4.3.3 共享存储器	98
3.4.2 并行性和功率	57	4.3.4 设备驱动器	100
3.4.3 多处理器设计方法的实际视点 ..	58	4.4 硬件互连机制	101
3.4.4 划分的形式	58	4.4.1 总线	101
3.4.5 处理器接口和互连	60	4.4.2 直接连接端口	104
3.4.6 任务间通信	63	4.4.3 数据队列	105
3.5 SOC 设计方法的新要素	63	4.4.4 时分复用处理器	108
3.5.1 SOC 设计流程	64	4.5 性能驱动的通信设计	108
3.5.2 新流程的基本步骤	65	4.5.1 复杂 SOC 设计中系统建模的 作用	108
3.6 解决 6 个问题	66	4.5.2 系统建模语言	109
3.6.1 提高 SOC 的可编程性	66	4.5.3 系统建模例子: XTMP	111
3.6.2 创建优化平台以积累产品产量 ..	66	4.5.4 平衡计算和通信	115
3.6.3 为取得供应优势使用可移植 IP 模块	67	4.6 SOC 设计流程	115
3.6.4 优化处理器的性能和效率	67	4.6.1 推荐的设计流程	115
3.6.5 用调整过的处理器代替硬连线 设计	68	4.6.2 SOC 设计方法中的改变	117
3.6.6 用以处理器为中心的 SOC 方法 统一硬件和软件	68	4.7 复杂 SOC 中的非处理器组成模块 ..	118
3.6.7 复杂 SOC 和 6 个问题	68	4.7.1 存储器	118
3.7 进一步的阅读资料	68	4.7.2 I/O 外围设备	119
第 4 章 复杂 SOC 的系统级设计	71	4.7.3 硬连线逻辑模块	120
4.1 复杂 SOC 系统架构的机遇	72	4.8 以处理器为中心的 SOC 架构的 影响	121
4.1.1 并行设计的基本过程	73	4.9 进一步的阅读资料	122
4.1.2 SOC——组件交互网络	73	第 5 章 可配置处理器: 软件视角	124
4.1.3 硅尺寸缩小对系统划分的影响 ..	75	5.1 处理器硬件/软件协同生成	125
4.1.4 为什么用多处理器	75	5.1.1 应用、编程语言和处理器 架构	125
4.1.5 并行性和系统架构的种类	76	5.1.2 小例子: 像素混合	126
4.1.6 迟滞、带宽和通信结构	78	5.2 指令定义和应用细调的过程	127
4.1.7 SOC 通信架构中的可靠性和 可缩放性	82	5.2.1 剖析和性能	128
4.1.8 通信编程灵活性	83	5.2.2 为了提高性能和效率的 新指令	131
4.1.9 交互机制的早期绑定和晚期 绑定	84	5.3 指令扩展的基础	131
4.2 以处理器为中心的 SOC 组织中的		5.3.1 指令扩展的方法	133
		5.3.2 应用的升级	137

5.3.3	指令集性能和通用性之间的 权衡	138	6.4.3	从微代码引擎到处理器 的转变	206
5.3.4	操作融合	139	6.4.4	微代码数据通路	208
5.3.5	复合操作	144	6.4.5	编码操作	209
5.3.6	SIMD 指令	145	6.4.6	微程序	211
5.4	程序员模型	147	6.5	设计处理器接口	212
5.4.1	基准用户指令集	148	6.5.1	存储器映射 RAM	214
5.4.2	专用指令集	150	6.5.2	存储器映射队列和寄存器	217
5.4.3	系统编程指令集	151	6.5.3	基于连线的输入输出	220
5.5	处理器性能因素	152	6.6	一个简单的例子: ATM 包分段和 重组	223
5.5.1	软件开发环境	155	6.7	处理器用于硬件替换的新功能	227
5.5.2	软件运行时环境	160	6.7.1	深理的任务引擎	227
5.5.3	处理器生成流程	162	6.7.2	带有备用处理器的设计	229
5.6	实例: 细调一个大型任务	163	6.7.3	系统监控处理器	230
5.7	存储器系统细调	167	6.8	处理器、硬件实现方法和验证 流程	231
5.7.1	基本的存储器系统策略	167	6.8.1	硬件流程	232
5.7.2	详细的存储器系统细调	168	6.8.2	验证流程	233
5.7.3	整体存储器系统性能	169	6.9	硬件抽象的发展	235
5.7.4	内部循环数据调用细调	171	6.10	进一步的阅读资料	235
5.8	长指令字	174	第 7 章	SOC 设计中的深入论题	237
5.8.1	代码长度和长指令	175	7.1	针对处理器性能的流水线设计	237
5.8.2	长指令字和自动处理器生成	177	7.2	深入处理器流水线阻塞	240
5.9	全自动指令集扩展	179	7.2.1	三种流水线阻塞类型	240
5.10	进一步的阅读资料	182	7.2.2	流水线和异常	243
第 6 章	可配置处理器: 硬件视角	184	7.2.3	对复杂指令的替代性流水线化	244
6.1	应用加速: 一个常见的问题	184	7.3	优化处理器来适应硬件	245
6.2	流水线和处理器简介	187	7.3.1	克服分支架构的差异	245
6.2.1	流水线的基本原理	187	7.3.2	克服存储器访问限制	250
6.2.2	RISC 流水线基础	187	7.4	多处理器调试和跟踪	250
6.2.3	扩展指令集实现的流水线	188	7.4.1	MP 调试	251
6.2.4	处理器硬件扩展正确性的 保证	190	7.4.2	MP 跟踪	252
6.3	从硬件模块到处理器	190	7.5	存储器系统中的问题	253
6.3.1	从硬件到指令的基本转换	191	7.5.1	多存储器端口流水线	253
6.3.2	每条指令一个原语操作	192	7.5.2	SIMD 指令集中的存储器对齐	254
6.3.3	每条指令多个独立操作	196	7.5.3	共享存储器的同步机制	255
6.3.4	流水线化的指令	199	7.5.4	指令 ROM	258
6.3.5	把硬件功能映射到处理器指令 的权衡	200	7.6	优化可扩展处理器的功耗	258
6.4	从硬连线引擎转移到处理器	201	7.6.1	内核功耗	259
6.4.1	把有限状态机转换成软件	201	7.6.2	可扩展能力对性能的影响	259
6.4.2	瞄准灵活性的专用处理器 设计	204	7.6.3	存储器功耗	260
			7.6.4	缓存器功耗指南	261