HZ BOOKS
华章教育

计 算 机 科 学 丛 书

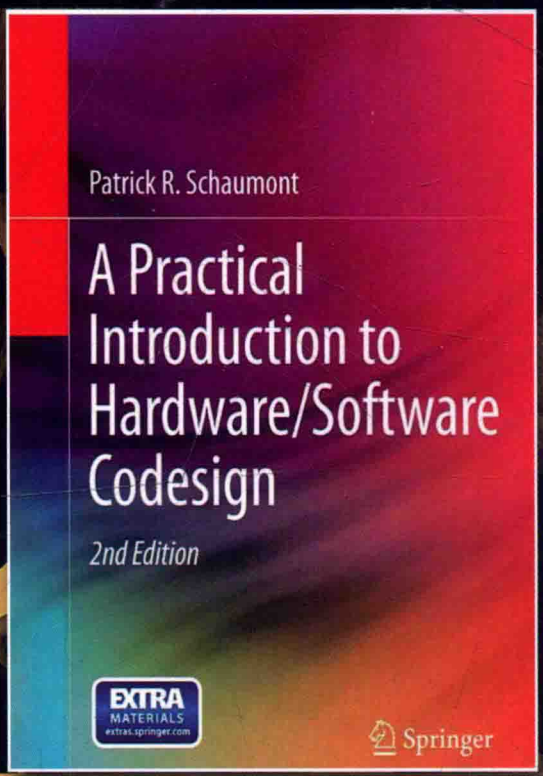
Springer

原书第2版

软/硬件协同设计

[美] 帕特里克 R. 肖蒙（Patrick R. Schaumont） 著
王奕 于恒 杨胜齐 哈亚军 译

A Practical Introduction to Hardware/Software Codesign
Second Edition



机械工业出版社
China Machine Press

计 学 丛 书

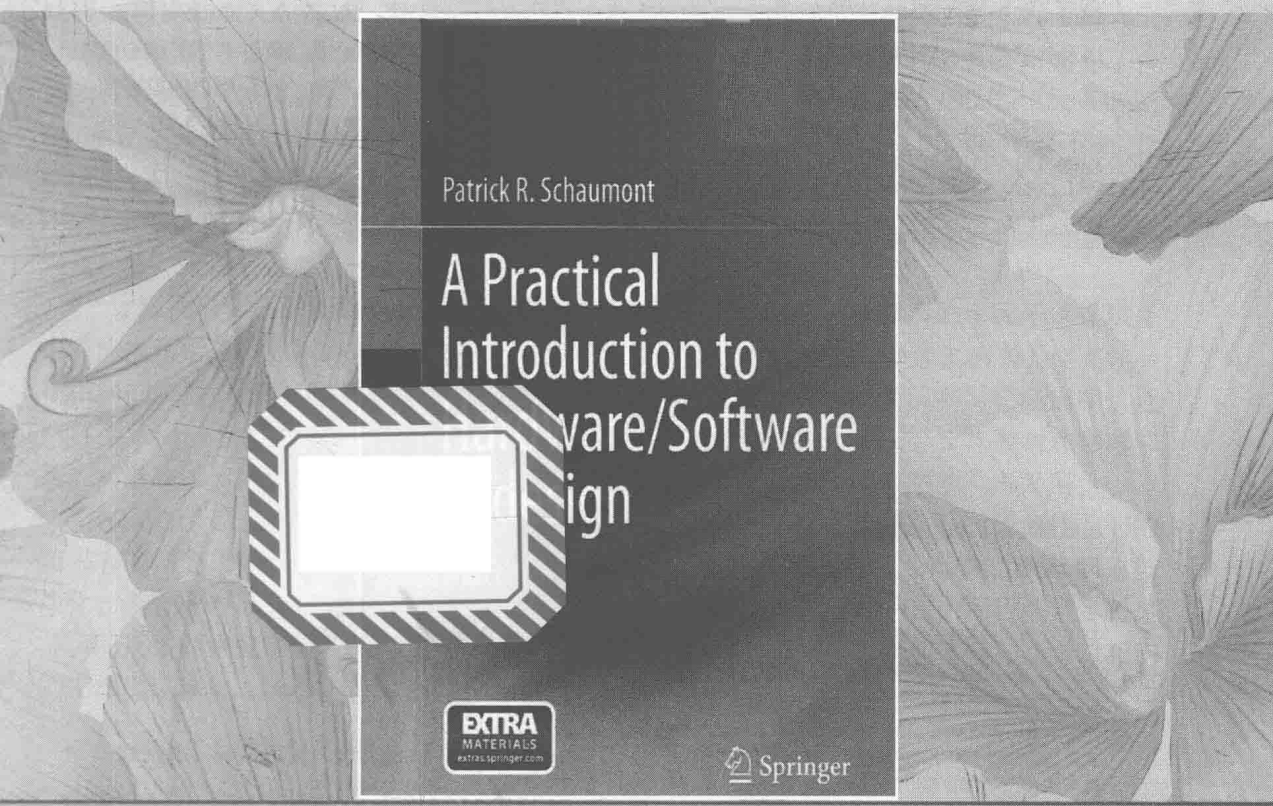
原书第2版

软/硬件协同设计

[美] 帕特里克 R. 肖蒙（Patrick R. Schaumont） 著
王奕 于恒 杨胜齐 哈亚军 译

A Practical Introduction to Hardware/Software Codesign

Second Edition



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

软/硬件协同设计（原书第2版）/（美）肖蒙（Schaumont，P. R.）著；王奕等译．—北京：机械工业出版社，2015.11
（计算机科学丛书）
书名原文：A Practical Introduction to Hardware/Software Codesign, Second Edition

ISBN 978-7-111-52018-4

I.软… II.①肖… ②王… III.电子计算机－系统设计 IV.TP302.1

中国版本图书馆CIP数据核字（2015）第266219号

本书版权登记号：图字：01-2014-2859

本书全面、深入地探讨软/硬件协同设计的4个主题：基础概念、自定义体系结构的设计空间、软/硬件接口和应用实例。首先介绍软/硬件的概念与性质、数据流系统的稳定性分析、将数据流模型实现为硬件和软件、带数据路径的有限状态机、微程序的系统结构、通用的嵌入式RISC内核，以及将通用嵌入式内核集成在片上系统（SoC）的FSMD模块中。其次描述软/硬件通信的核心概念、片上总线的结构、微处理器接口，以及把硬件模块封装到一个预定义的软/硬件接口的设计技术。最后给出3个软/硬件协同设计的应用实例，涉及Trivium流密码算法协处理器、AES协处理器以及CORDIC协处理器。

本书是软/硬件协同设计方面的经典图书，内容丰富，适合作为高等院校计算机、通信、电子电工等相关专业本科生及研究生的教材，也是广大技术人员的极佳参考读物。

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：张国强　　责任校对：董纪丽
印　　刷：中国电影出版社印刷厂　　版　　次：2016年1月第1版第1次印刷
开　　本：185mm×260mm　1/16　　印　　张：22.75
书　　号：ISBN 978-7-111-52018-4　　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　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

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。

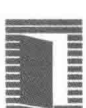
近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

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，Brain 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，我们的联系方法如下：

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

高效的电子系统设计是否有一定之规？软件设计者常会说：把代码写得精益求精！而硬件专家常会说：让电路运行得更快！但你看过本书之后，将得出显而易见的结论。只有兼收并蓄，把软件与硬件设计的长处融合为一，才有望达到最佳效果。软 / 硬件协同设计就是这样的一门学问，它引导设计者在电子系统设计中针对性能与设计灵活性进行明智取舍。通过进行软 / 硬件协同设计，设计者可对两种区别巨大的设计风格——使用软件对功能在时间维度进行串行分解和使用硬件对功能在空间维度进行并行分解——加以融合，从而得到满意的设计。

帕纳马伦科的奇幻飞行器

图 1 中的手稿描绘了一个有趣的人力飞行器，该飞行器由比利时艺术家帕纳马伦科于 1973 年设计，名字为 Meganeudon II。据我所知，尽管它从未被成功设计，但我相信这份手稿已经充分描述了帕纳马伦科的设计概念。事实上，设计并不在乎复杂度，也不关乎底层细节。设计所要表达的只关乎想法、概念、视野。设计本质上就是一种创造的过程。

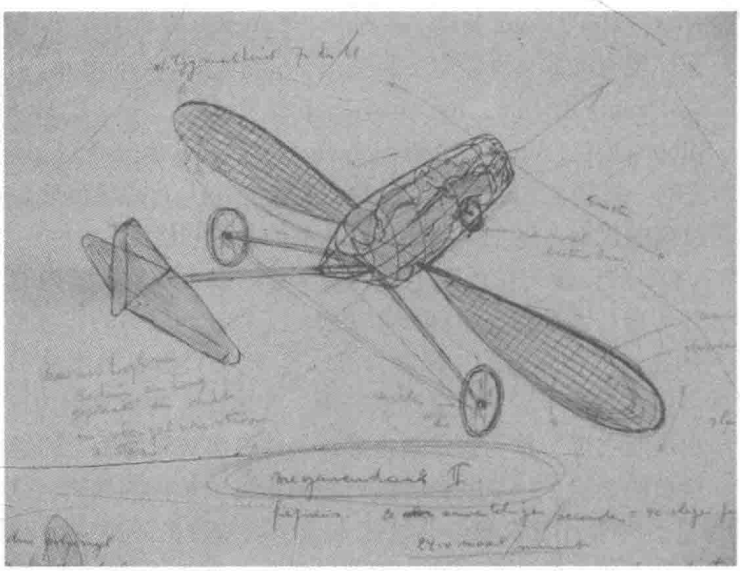


图　1

然而，要实现一种设计，我们需要技术，这样才能将想法与草稿投射到真材实料。幸运的是，作为计算机工程师，我们总是能够近水楼台先得月。一切皆因对技术的信手拈来，我们才可以将梦想照进现实。

本书读者对象

阅读本书之前，你应该基本了解硬件的相关知识，熟悉诸如寄存器、逻辑门、多路复用

器、算术运算电路等电子元器件的概念。你也应该了解如何运用C语言编写代码。这些知识通常在计算机工程系的入门课程中学习，或者通过学习数字系统设计和软件工程两门课程进行训练。

本书适合作为高年级本科或低年级研究生的教材。另外，它也适合非计算机工程专业背景的研究者阅读，例如，无系统硬件设计训练背景的密码学家可通过研习本书，从而达到针对特殊算法设计系统架构的程度。

内容组织

本书重点兼顾设计方法与设计语言建模。设计建模有助于帮助设计者厘清问题头绪，捕捉问题答案。设计方法则系统性地将设计模型转换为最终实现。

本书包含四部分：基本概念、自定义体系结构的设计空间、软/硬件接口，以及应用实例。

第一部分　基本概念

第1章介绍软件和硬件的基本性质，并讨论软/硬件协同设计的动因。第2～3章描述数据流建模与实现。数据流建模是非常有用的系统级规范技术，并且是实现独立的，即系统可以从数据流模型实现成软件或者硬件。数据流建模也支持高层次性能分析与优化。第2章着重介绍数据流系统的稳定性分析与数据流模型的性能优化策略，例如流水线与重定时。第3章展示如何将数据流模型实现为硬件或软件。第4章介绍C代码的控制流和数据流分析。通过分析一段C代码中的控制依赖与数据依赖，设计者可以洞悉该程序的可能硬件实现。

第二部分　自定义体系结构的设计空间

第二部分带领读者开启一次灵活的、自定义体系结构的广阔设计空间之旅。四种数字体系结构的回顾展现了硬件设计如何逐步演化成软件设计。第5章介绍的带数据路径的有限状态机（FSMD）是整个“旅途”的起始点。有限状态机模型与寄存器传输级（RTL）的硬件模型等价。第6章介绍微程序的体系结构。与RTL机器不同之处在于，它们具有一个软件可编程的灵活控制器。第7章回顾通用的嵌入式RISC内核。这些处理器是当代软/硬件系统的核心部件。第8章介绍如何将这些通用嵌入式内核集成到片上系统（SoC）的FSMD模块中。第三部分将讨论针对片上系统的软/硬件协同设计的问题。

第三部分　软/硬件接口

第三部分描述片上系统（SoC）中硬件与软件的交互机制。第9章介绍软/硬件通信的核心概念。其中详细解释了软/硬件的同步策略，以及通信限制设计与计算限制设计的区别。第10章讨论片上总线的结构，以及软件与硬件通过片上总线高效交互的若干技术。第11章描述微处理器接口。这些接口的主要作用是为外设硬件在基于处理器的系统设计中提供一个处理器的附着点。该章描述三种接口，分别为内存映射接口、协处理器接口和自定义指令接口。第12章讨论了把硬件模块封装到一个预定义的微处理器接口的设计技术。

第四部分　应用实例

第四部分描述三个软/硬件协同设计的应用实例。第13章给出用于Trivium流密码算法

的一个协处理器设计方案。第 14 章给出用于 AES 的一个协处理器设计方案。第 15 章给出用于计算 CORDIC 旋转的一个协处理器设计方案。每个设计方案都涉及不同的处理器和微处理器接口。第 13 章使用 8051 微控制器和 ARM，第 14 章使用 ARM 和 Nios-II，第 15 章使用 MicroBlaze。

本书大部分实例都可从互联网下载，供读者动手练习。附录包含 GEZEL 工具与示例的安装指导。

每章末尾两节分别包含问答题和扩展阅读。问答题用于帮助读者建立对本书内容更深层的理解。部分问答题的答案可通过 Springerextras 在线获取，网址为 http://extras.springers.com。[⊖]

有些主题本书并未提及与讨论。本书作为一本关于一项复杂主题的参考指南，试图把握细节度与复杂度之间的平衡。例如，本书并未涉及软件并发的进阶概念，如线程和复杂软件架构（如操作系统、驱动程序等）。本书也未涉及软件中断或者更高阶的软件系统操作概念，如 DMA（Direct Memory Access）。

建议读者顺序阅读本书，必读内容包含第 1、4、5、7 ～ 12 章。

第 2 版出版说明

第 2 版全面修订了第 1 版，重新撰写了几个章节并增添了新内容。第 2 版的重点放在提高整体结构，使其更加有逻辑性和可读性，并添加了一些应用实例。

以下是具体的修订变化：

- 数据流的相关章节被划分为两章：一章着重介绍数据流的分析和传输，另一章着重介绍数据流的实现。针对传输的讨论为介绍本书的性能分析和优化提供了条件。
- 第 6 章新增了一个基于 8051 微控制器的微程序实例。
- 第 7 章在 RISC 处理器的基础上，着重于通过 GNC 编译器的工具链检验目标代码以及分析汇编代码。
- 由于 GEZEL 开始支持 AVR 指令的仿真，第 8 章在片上系统中新增了一个使用 AVR 微处理器的应用实例。
- 第三部分重新组织了描述软 / 硬件接口的内容。第 9 章解释了软 / 硬件接口设计中的类属概念。在第 1 版中，这些概念在各个不同章节中分散介绍。第 2 版将这些概念集中在一章内，以便给读者一个更加简明的定义。
- 第三部分对软 / 硬件接口的三个组成部分进行了详细说明。这三个部分包括片上总线（第 10 章）、微处理器接口（第 11 章）以及硬件接口（第 12 章）。“硬件接口”在第 1 版中叫“控制器”(Control Shell)。新的定义在逻辑上更加适合软 / 硬件接口的整体描述。
- 第 10 章在片上总线内容的基础上新增了对 Altera 的 Avalon 片上总线的描述。AMBA 的部分也更新到了最新的 AMBA 规范标准（v4）。
- 第 11 章在微处理器接口内容的基础上新增了 Nios-II 自定义指令接口的讨论和实例。
- 第四部分在原有应用实例内容的基础上扩展了一个 AES 协处理器的章节。现在的内

⊖ 关于本书资源，读者可与施普林格亚洲有限公司北京代表处联系，电话：010-82670211-895，电子邮件：parick.chen@springer.com。——编辑注

容包括三章，涉及 Trivium、AES 以及 CORDIC。

- 新增的附录描述如何安装和使用 GEZEL 工具。现在第 5、6、8、11、13 ～ 15 章的实例有可用的源代码。使用 GEZEL 工具可编译这些源代码。具体的操作步骤可以参考附录。
- Springer 网站的 extras 部分提供了部分问答题的答案。
- 第 2 版全面修订了语法和录入错误。非常感谢 Gilberta Fernandes Marchioro、Ingrid Verbauwhede、Soyfan 和 Li Xin 提供了第 1 版中的勘误。

实例演练

本书既着眼于概念与设计方法，又兼顾实践联系，因而在章节之间穿插了很多实例，并且使用 3 章的篇幅（即第四部分）详细讨论了完整的设计流程。

本书选择 GEZEL（一种开源周期精准的硬件建模语言）作为本书使用的硬件描述语言。可从 GEZEL 网站（http://rijndael.ece.vt.edu/gezel2）下载软件、用例，以及其他说明文档。请参见附录 A 中的下载细节与安装说明。

之所以选择 GEZEL 而非主流硬件描述语言（HDL）如 VHDL、Verilog 或 SystemC，原因如下：

- 减少建模开销。尽管建模对于嵌入式系统的构建至关重要，但是太多的建模细节往往使读者忽略掉关键问题。例如，对时钟信号建模需要很多额外的细节，但对单时钟源同步系统的设计来说，它们又不重要。在现实中，数字硬件系统大部分由单时钟源同步系统构成。
- GEZEL 可安装协同仿真插件。GEZEL 模型可与多种处理器仿真模型进行协同仿真，包括 ARM、8051 和 AVR。GEZEL 包含一种库模块建模机制，允许用户定义与多种仿真引擎进行协同仿真的接口。
- 简洁化的需求。本书侧重实践，因而会列举大量实例代码清单，应力求代码简洁。第 5 章中同时罗列了用 GEZEL、VHDL、Verilog、SystemC 描述的设计实例，帮助读者进一步明晰简洁的重要性。
- 实现路径。GEZEL 可自动转换为 VHDL，因而可用标准 HDL 逻辑合成工具生成系统。

作者曾在软 / 硬件协同设计课程上把本书作为课程辅导材料。课程受众包括高年级本科生与低年级研究生。对于高年级本科生，这门课程将计算机工程的多个元素混合在一起，包括计算机体系结构、软件工程、硬件设计、调试以及测试。对于低年级研究生，这门课程可作为他们重温知识以及开始计算机科研生涯的起点。

在软 / 硬件协同设计课程中，GEZEL 实验由 FPGA 后端（基于 Xilinx/EDK 或者 Altera/Quartus）与 FPGA 快速原型套件实现。GEZEL 建模作业与 FPGA 实现作业交叉进行。通过 GEZEL 后端套件进行实现，学生甚至不必进行 VHDL 编码。在课程最后有一个竞赛环节。学生需将一段指定的 C 程序通过软 / 硬件协同设计技术在 FPGA 上实现尽量快加速。

致谢

衷心感谢为本书的出版做出贡献的那些人。

家庭在我生命中是永恒的后盾。非常感谢家庭成员的耐心、鼓励和热情。他们的价值与真诚使我时刻受到鼓舞。

在与很多出色的工程师的交流中，我产生了撰写本书的想法。我的博士生导师鲁文大学的 Ingrid Vebauwhede 教授从一开始就支持 GEZEL 在研究和教学中的应用。她也是把本书用作教材的第一批教师之一。丹麦工业大学的 Jan Madsan 教授、加州大学欧文分校的 Frank Vahid 教授是我学习的教育模范。每次与他们的讨论都是对我的一种鼓励。

在本书第 1 版出版后，我也与教授协同设计的很多老师交换了意见。非常感谢 Jim Plusquellic（新墨西哥大学）、Edwad Lee（加州大学伯克利分校）、Anand Raghunathan（普渡大学）以及 Axel Jantsch（瑞典皇家理工学院）。感谢斯普林格的 Chuck Glaser，是他鼓励我撰写本书第 2 版，并且给予我很多有深刻见解的有用建议。感谢 Grant Martin（Tensilica）为本书写了评论。

通过多年开发，现今 GEZEL 已经得到越来越多用户的认可。这些用户非常有耐心，尽管他们遇到过很多故障，可还是成功实现了协同设计项目。这些用户包括：Aske Brekling、Herwin Chan、Doris Ching、Zhimin Chen、Junfeng Fan、Xu Guo、Srikrishna Iyer、Miroslav Knezevic、Boris Koepf、Bocheng Lai、Yusuke Matsuoka、Kazuo Sakiyama、Eric Simpson、Oreste Villa、Shenling Yang、Jingyao Zhang 等。我相信还有很多其他的人，如果漏掉了任何人，我表示抱歉。

最后，衷心感谢弗吉尼亚理工大学选择协同设计课程（ECE4530）的学生。每年，我都从他们身上学到很多。我对他们提出的问题和想法及归纳的结论感到钦佩。他们是工程师，但是我可以说，他们中的一些也是艺术家。

希望你会喜爱本书，并且衷心地希望本书在实际设计中能帮到你。我为本书中还存在的一些错误感到抱歉，当然我感谢你对本书的反馈。

Patrick R. Schaumont

美国　弗吉尼亚州　布莱克斯堡

第四部分 应用实例

第一部分

A Practical Introduction to Hardware/Software Codesign, Second Edition

基本概念

在本书的第一部分介绍软/硬件协同设计的重要概念。我们比较和对照了电子设计领域的两种风格：一是以硬件设计为主导的思路，二是以软件设计为主导的思路。所谓的软/硬件协同设计并不是简单地把软件与硬件组件胶合在一起，事实上，它需要综合考量两种设计的特点并做出恰当的选择，以实现系统的灵活性与性能的平衡。

并行和顺序实现之间的权衡是软/硬件协同设计者的另一个基本问题。我们将讨论一个并发系统模型（数据流），它既可以转换成以硬件并行实现，也可以以软件顺序实现。

最后，我们将展示如何分析一个C程序的代码，并把它分解成控制流和数据流。这种分析对于理解如何迁移C代码到硬件是至关重要的。在接下来的讨论中会揭示，软/硬件协同设计的一个通常做法是将软件提供的功能迁移到硬件上进行实现，从而提高应用的整体性能。

第 1 章

A Practical Introduction to Hardware/Software Codesign, Second Edition

何为硬件，何为软件

1.1 软 / 硬件协同设计简介

在电子系统设计中，软 / 硬件协同设计是一个包容万象的概念。为了有助于不同背景的读者拥有统一的概念，本节给出一个针对硬件和软件的简要定义。本节还通过一个精简的小例子让大家初步了解软 / 硬件协同设计，并在最后给出软 / 硬件协同设计的定义。

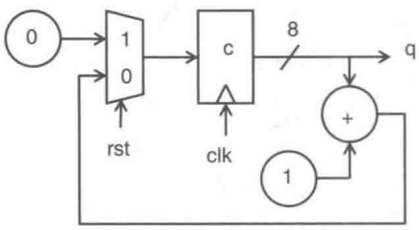
1.1.1 硬件

在本书中，我们将硬件建模为由组合逻辑和触发器组成的单时钟同步电路。该单时钟同步电路由基本模块组成，如寄存器、加法器和乘法器等。电路级行为可以认为是逻辑和算法操作信号在寄存器间的时序变换，因此由时钟驱动的硬件模型通常叫作寄存器传输级（Register-Transfer-Level，RTL）模型。

图 1-1a 给出了一个在寄存器传输级（RTL）上以硬件实现的计数器实例。控制信号 rst 决定着寄存器的值应该清零还是保持不变。在时钟信号 clk 的上升沿，寄存器 c 的输出端值即 q 值将被更新。在图 1-1a 中，除 rst 和 clk 信号之外，其他单线的位宽为 8 位。除图 1-1a 中用直观的图形来描述硬件外，本书中我们还将使用硬件描述语言 GEZEL 来等价描述。例如，图 1-1b 展示了使用 GEZEL 来描述图 1-1a 所示的计数器。在第 5 章，我们会更详细地介绍 GEZEL 语言。

图 1-1c 使用时序图来描述硬件的行为。其中横坐标代表时间轴自左向右，纵坐标罗列电路中的不同信号。在图 1-1c 中，由于 rst 清零正脉冲的作用，寄存器输出端 q 在时钟沿 2 被清零，而从时钟沿 3 开始，rst 位于低电平，因而输出端 q 值得以累加。时钟沿 2 之前的阴影部分表示 q 的赋值是未知的。相对于通过图形或者 GEZEL 静态地来描述电路，时序图则辅助我们更清楚地读懂电路各组件之间的逻辑行为关联。本书中，我们将用时序图来描述软件 – 硬件接口的底层电路行为，以及片上总线的信号变化。

另外，尽管单时钟同步电路在建模更加复杂的电路设计（如异步电路、动态逻辑、多相时钟电路、含锁存器的电路等）时存在局限性，但因其简洁性，读者可以一目了然每一个时钟下电路的信号变化，所以本书中我们采用它为硬件建模，以便解释软 / 硬件协同设计中的重点概念。

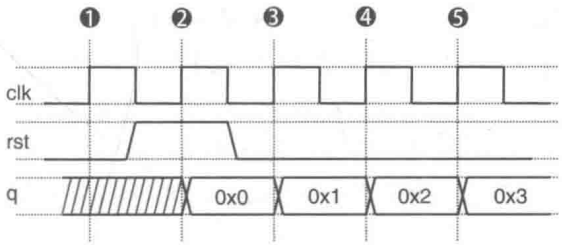


a）计数器的硬件模块表示

```
dp count(in  rst : ns(1);
         out q   : ns(8)) {
  reg c : ns(8);
  always {
    q = c;
    c = rst ? 0 : c + 1;
  }
}
```

b）计数器的等价 GEZEL 描述

图 1-1　在寄存器传输级（RTL）上以硬件实现的计数器实例



c）时序图

图 1-1 （续）

1.1.2 软件

软 / 硬件协同设计问题中，软 / 硬件接口的实现，尤其是底层软件架构的细节相当重要，因为这直接影响设计的性能和软 / 硬件接口的实现成本。本书将讨论软件底层的重要实施环节，如变量在内存中的组织形式，以及如何从高级编程语言（如 C）中控制底层实施的一些技术细节。

这里，我们把“软件”定义为单线程顺序执行的程序，可用 C 语言或者汇编语言实现。代码清单 1-1 和代码清单 1-2 分别给出了以 C 语言和汇编语言实现的求数组最大值的范例。注意，本书中讨论的大部分软件实现与处理器架构无关。在某些情况下，我们假定软件运行在一个 32 位架构的处理器（如 ARM）或者一个 8 位架构的微控制器上（如 8051）。

代码清单1-1　C范例

```
1  int max;
2
3  int findmax(int a[10]) {
4    unsigned i;
5    max = a[0];
6    for (i=1; i<10; i++)
7      if (a[i] > max) max = a[i];
8  }
```

代码清单1-2　汇编范例

```
        .text
findmax:
        ldr     r2, .L10
        ldr     r3, [r0, #0]
        str     r3, [r2, #0]
        mov     ip, #1
.L7:
        ldr     r1, [r0, ip, asl #2]
        ldr     r3, [r2, #0]
        add     ip, ip, #1
        cmp     r1, r3
        strgt   r1, [r2, #0]
        cmp     ip, #9
        movhi   pc, lr
        b       .L7
.L11:
        .align  2
.L10:
        .word   max
```