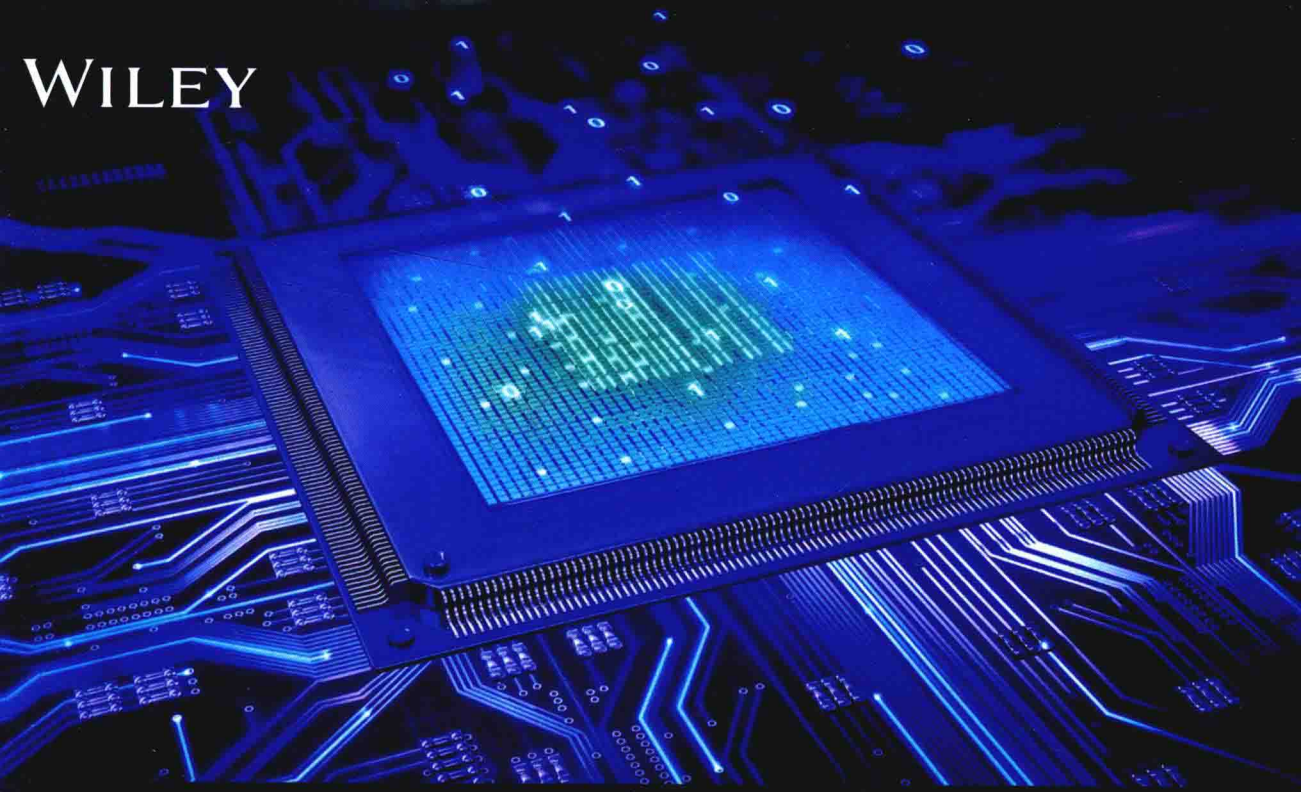
WILEY



Learning Computer Architecture with Raspberry Pi

使用Raspberry Pi学习计算机体系结构

Eben Upton Jeff Duntemann
Ralph Roberts Tim Mamtora 著
Ben Everard
张龙杰 杨玫 孙涛 于亮 译

清华大学出版社

使用Raspberry Pi学习计算机体系结构

Eben Upton
Jeff Duntemann
[美] Ralph Roberts 著
Tim Mamtora
Ben Everard

张龙杰 杨玫 孙涛 于亮 译

清华大学出版社
北 京

Eben Upton, Jeff Duntemann, Ralph Roberts, Tim Mamtora, and Ben Everard
Learning Computer Architecture with Raspberry Pi
EISBN：978-1-119-18393-8

图书在版编目(CIP)数据

使用Raspberry Pi学习计算机体系结构/(美)艾本·阿普顿(Eben Upton) 等著；张龙杰 等译. —北京：清华大学出版社，2018
书名原文：Learning Computer Architecture with Raspberry Pi
ISBN 978-7-302-48717-3

Ⅰ. ①使… Ⅱ. ①艾… ②张… Ⅲ. ①Linux操作系统 Ⅳ. ①TP316.85

中国版本图书馆 CIP 数据核字(2017)第 271338 号

责任编辑：王　军　李维杰
封面设计：周晓亮
版式设计：方加青
责任校对：曹　阳
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：25.25　　字　　数：567 千字
版　　次：2018 年 1 月第 1 版　　印　　次：2018 年 1 月第 1 次印刷
印　　数：1 ～ 4000
定　　价：79.80 元

产品编号：069499-01

译者序

大师出手，必是精品。本书的首席作者Eben Upton正是Raspberry Pi的设计者之一，也是Raspberry Pi基金会的创始人。

Raspberry Pi的出现以及Raspberry Pi基金会的创立最初都源自一个简单的目的：推动青少年和成人教育，尤其是在计算机、计算机科学以及相关学科领域。作者Eben Upton在英国剑桥大学负责计算机教学管理工作期间，敏锐地觉察到现代年轻人对计算机的兴趣持续降低，最直接的表现是申请大学计算机科学专业的学生数量逐年下降。为了改变这种趋势，同时为院校计算机教学活动的开展提供一个低成本的通用平台，Eben Upton及其团队联合设计出Raspberry Pi这块神奇的小板子。自2012年上市以来，短短5年时间，Raspberry Pi就从1代发展到3代，并在全世界范围内取得了巨大成功。如果说20世纪80年代Commodore公司生产的低成本可编程计算机引发了一场技术革命，那么如今，随着Raspberry Pi的出现，计算领域的第二场革命正在孕育中。

对于广大民众而言，一方面是计算机行业如火如荼地发展，各种计算机应用层出不穷，乱花纷呈渐迷人眼；另一方面却是底层核心技术的层层封装，日渐隐匿在各种计算机变体的华丽外衣之下，执子之手却隔千年。《使用Raspberry Pi学习计算机体系结构》正是在这种背景下出现的，其中心目的只有一个：帮助读者建立起对计算机工作过程的系统认识。

作者以一种夹叙夹议的口吻完成整部作品。讨论技术的同时，追根溯源，层层递进，为读者呈现了一部计算机发展演变的历史剧。全书共12章，内容涵盖CPU和SoC、电子和非易失性存储器、各种有线及无线网络、程序设计、操作系统、视频编码和压缩、3D图形处理、语音处理、输入/输出接口等。在对各个环节的讲解中，作者始终坚持理论结合实际，将所有的核心内容通过Raspberry Pi呈现出来，这也是本书的一大特色。

第1章主要介绍Raspberry Pi，包括Raspberry Pi的前世今生、主要功能模块和各种接口。

第2章主要介绍计算的相关概念。作者从公元前的算盘开始，以高度概括的语言带领我们开启了一趟跨时空旅行：算盘、机械式计算器、电动机械式计算机、可编程计算机……。在此基础上，进一步讲解程序的本质、存储器的作用、系统总线的功能、操作系

统的出现以及二进制、十六进制等计算机基础知识。

第3章围绕电子存储器这个主题展开。同样的讲解风格，由历史开始，旋转磁存储器、磁芯存储器、SRAM、DRAM、缓存、虚拟存储器，当然最后以Raspberry Pi上的存储器收官。本章以硬件讲解为主，在存储器硬件结构的基础上，分析其工作原理和特性。

第4章主要讲解CPU和SoC。作者首先以搭积木的方式，从构成CPU最基本的逻辑器件开始，由晶体管到触发器，再由触发器到加法器、移位寄存器。随后介绍了栈、时序、CISC、RISC、中断、流水线技术等知识。最后以ARM11内核作为具体对象，进一步讲解协处理器、指令的顺序和乱序执行，以及SoC等知识。

第5章围绕程序设计展开讨论。作者围绕软件开发过程、汇编语言、高级语言、程序编译、面向对象程序设计等几个主要话题，对计算机程序设计的各个方面进行了详细的阐述。

第6章主要讲解非易失性存储器。作者从非易失性存储器的历史开始，介绍打孔卡、磁带、软盘、光盘、硬盘、Flash存储器的起落兴衰。围绕光盘、硬盘、Flash存储器等主流非易失性存储器，进一步讲解其硬件构成、工作原理和读写特性。

第7章以网络作为讨论的中心话题，介绍网络的基本协议和工作过程，重点对以太网和Wi-Fi的核心技术和常见问题进行了剖析。

第8章主要介绍操作系统。作者对操作系统的主要功能——进程管理、存储器管理、文件系统管理、设备管理——逐一剖析，最后概要介绍了Raspberry Pi上的操作系统。

第9章涉及一个较新的领域：视频编解码和压缩。作者详细介绍了视频的编码原理和视频压缩算法背后对人体视觉生理的考虑。在此基础上，进一步介绍了各种视频编码、解码标准。

第10章围绕3D图形展开讨论。使用大量的篇幅系统介绍了应用广泛的OpenGL、OpenVG的有关知识。本章内容涉及较多的计算机图形学方面的知识，需要读者具备一定的基础。

第11章主要介绍语音。作者围绕如何使计算机播放更逼真的声音这个话题展开讨论，介绍语音的编码和解码方法，并以Raspberry Pi平台为例，介绍不同语音接口的特性和高品质语音的获取方法。

第12章主要介绍各类计算机输入/输出设备和接口。作者基于Raspberry Pi平台，重点介绍了USB、SCSI、PATA、SATA、RS-232、HDMI、I2S、I2C以及GPIO。

计算机是一个有序整体，而不是各种电路板的简单堆砌物。本书有助于帮助广大计算机爱好者(特别是那些拥有Raspberry Pi的读者)建立完整的计算机系统观。既可以作为Raspberry Pi爱好者的辅助学习资料，也可作为计算机体系结构等课程的参考教材，还可作为计算机相关专业学生的自学教材。

本书所有章节由张龙杰、杨玫、孙涛、于亮翻译，并由张龙杰统稿。参与本次翻译的还有卢祯烨、程露露、陈青华、刘家祺、庞伟、樊鹏飞、孙海文。

特别感谢计算机硬件爱好者卢祯烨。卢祯烨对原著爱不释手，并在学习原著的过程

中，参与了部分硬件内容的初稿翻译工作。在本书中文版出版之际，卢祯烨也将赴美国继续从事计算机硬件方面的学习和深造，祝福他在该领域取得更好的成绩。

在翻译过程中，我们力求在忠于原文的前提下尽量以适合国人的阅读风格再现原书风貌，但由于译者水平有限，加之本书原著信息含量很大，内容涉猎广泛，对原文的理解和中文表达定有不当之处，译稿中难免有疏漏甚至错误，敬请读者批评指正。读者可将意见发送到ljcx1213@163.com，以求进一步提高译稿质量。

张龙杰

作者简介

Eben Upton是Raspberry Pi基金会的创始人，也是该基金会旗下的贸易部门Raspberry Pi(贸易)有限公司的CEO，他与Gareth Halfacree合著了*Raspberry Pi User Guide*一书。Eben早年创办了两家成功的移动游戏和中间件公司(Ideaworks 3d 和Podfun)，还曾担任剑桥大学圣约翰学院计算机科学的教学主管，并和他的父亲Clive Upton教授一起编写了牛津诗韵词典(Oxford Rhyming Dictionary)。Eben在剑桥大学获得了物理学和工程学学士学位以及计算机科学博士学位，还获得了工商管理硕士学位。

Jeff Duntemann从1974年就开始出版各种技术类和科幻类出版物。它是Xerox公司的程序师，同时还担任Ziff-Davis出版社和Borland国际软件公司的技术编辑。他编辑发行了两份程序师杂志，在其名下有20本技术性书籍，包括最畅销的*Assembly Language Step By Step*一书。在*Dr. Dobb's Journal*杂志上，Jeff连续四年撰写“Structured Programming”专栏，并在很多杂志上发表了大量技术性文章。1989年，Jeff和他的作家伙伴Keith Weiskamp发起成立了Coriolis Group，到1998年Coriolis Group已经成为美国亚利桑那州最大的图书出版商。Jeff对“强”人工智能表现出持久的兴趣，他的绝大部分科幻著作(包括*The Cunning Blood*和*Ten Gentle Opportunities*两本小说)都在探寻强人工智能的因果逻辑。Jeff的其他兴趣包括望远镜和风筝，他还是一位电子学和无线电业余爱好者(呼号K7JPD)。在过去的40年里，Jeff一直和妻子Carol居住在美国亚利桑那州的菲尼克斯市，陪伴他们的还有四条卷毛比雄犬。

Ralph Roberts是一名受过嘉奖的越战老兵，在阿波罗登月工程期间供职于NASA。自从1979年在*Creative Computing*杂志上发表第一篇文章开始，Roberts就一直从事计算机和软件方面的写作。Roberts为国家出版商撰写了超过100本书籍，以及上千篇文章和短篇小说。总而言之，他已经发售了超过2000万字的专业内容。Roberts的最佳畅销书包括美国第一本关于计算机病毒(美国国家电台由此诞生了多部相关电影)的书籍，还有过去21年里反复印刷的一本烹饪食谱——*Classic Cooking with Coca-Cola*，目前已经售出了50万本。

Tim Mamtora是博通有限公司IC设计部门的总工程师，目前是美国GPU硬件团队的技术带头人。他在移动计算机图形学方面从事了近七年的工作，此前为模拟电视和传统DSP

硬件开发内部IP。Tim拥有剑桥大学工程学硕士学位，其中第三年在马萨诸塞州技术学院度过，在那里激发了Tim对数字硬件设计的兴趣。他对推进工程充满激情，并专门花费时间在剑桥大学指导学生，他还在母校发表工程学机遇方面的演讲。工作之余，Tim喜欢各类体育运动、摄影以及游览世界。

Ben Everard是一位作家，也是一位播客。平时为Linux编写修补代码，还喜欢摆弄机器人。本书是Everard的第二部著作，他还撰写了*Learning Python with Raspberry Pi*一书(Wiley出版社，2014)。可以在推特@ben_everard上找到他。

技术编辑简介

Omer Kilic是一位嵌入式系统工程师，喜欢研究各式各样的小型互联计算机。他工作于硬件和软件工程实践、产品研发以及制造业的各种交集中。

前　言

在我10岁的时候，我的一位老师让我安坐在学校的一台计算机前。但不是你想象的那样，他没有给我讲解神秘的计算机程序设计知识，尽管它是BBC Micro(英国编程设计最容易、架构最复杂的8位微型计算机，在这台计算机上我借助BASIC和汇编语言成长起来)。相反，此前在学术兴趣、业余爱好和理想抱负之间，我面临一道多项选择难题，而这台神奇的机器为我理想的未来职业做出了判决：微电子芯片设计师。

听起来有点匪夷所思，不仅仅因为我希望成为一名计算机游戏程序师(好吧，是宇航员)，还因为在我的直接生活环境中并没有人有意识地为一个10岁的孩子设计一条通向微电子芯片设计的阳光大道。随后的几年里，我在学校学习了不少数学和科学知识，在家里学习程序(游戏)设计——最初在BBC Micro上，随后转到Commodore Amiga上。在这个过程中不时地——虽然不是很成功——涉猎到电子学领域。结果就是，运气超越了客观选择，我误打误撞地走上了一条通向最终目标的看似合理的道路。不过，直到18岁进入剑桥大学那年，我才真正明白自己认识上的差距在哪里。

剑　桥

在计算机科学史上，特别是在实践或应用计算发展史上，剑桥大学占据着特殊的地位。20世纪30年代末，年轻的剑桥大学学者阿兰·图灵(Alan Turing)证明停机问题(即“这个计算机程序是否会结束或停机”的问题)是不可计算的。或者从本质上讲，无法编写这样一个计算机程序，它能够分析其他任意的计算机程序并确定其是否会停止。在同一时间内，阿隆佐·邱奇(Alonzo Church)得出了相同的结论。他们现在共享这一成果：邱奇-图灵论题(Church-Turing thesis)。不过，邱奇采用纯数学方法完成了他的证明，图灵则基于递归函数，以序列操作的形式完成计算，也就是我们现在所谓的图灵机：机器头在无限长的纸带上来回移动，读取符号信息，改变自己的内部状态和移动方向，并写下新的符号。尽管绝大多数这类机器都是用于专门研究目的的，但是图灵引入了虚拟机的概念，通过写在纸带上的命令配置虚拟机，以此模拟其他任何具有特殊目的的机器的行为。这是如今这个普

通概念——通用可编程计算机——首次出现。

第二次世界大战爆发后，图灵继续在位于布莱切利园(Bletchley Park)开展的盟军代码破解工作中发挥了核心作用。在这个过程中，图灵(作为团队一员——别相信在电影中看到的)卷入到大量专用硬件的研发工作中，包括机电炸弹机，这台机器加速了破解德国恩尼格码密码的自动化进程。这些设备都没有使用图灵初始实验想法中的“有限状态机加无限纸带”这种具体架构，但与实际执行相比，却更适于数学分析。不过，即使是纯粹的电子巨人Colossus——如同炸弹机处理恩尼格码一样处理繁复冗杂的洛伦兹流密码——也没有触及通用程序设计的边界。尽管如此，对于这一代理论工程师而言，当他们返回到平民生活以后，使用真空管研发用于代码破解、雷达和火炮以及实现数字逻辑电路的大规模电子系统的经历，使得他们极具革新能力。

在莫里斯·威尔克斯(Maurice Wilkes)的带领下，位于剑桥大学数学实验室的一个工程师小组着手搭建了电子延迟存储自动计算机(Electronic Delay Storage Automatic Computer，EDSAC)。1949年投入运行时，时钟频率达到500KHz，通过两个温度控制水箱中的32条水印延迟线，构成了一个容量为2KB的易失性存储器。程序和代码可以通过纸带读写。美国和英国的很多机构都可以狭隘地宣称自己首次研发出通用数字计算机，仅仅是标榜“首次”。对于EDSAC，公开的声明指出，它是第一台将应用扩展到研发团队之外的计算机。其他学科的学者可以申请时间在这台机器上运行他们自己的程序，并由此引入了以计算为服务的概念。EDSAC之后研发了EDSAC II，然后是泰坦(Titan)。直到20世纪60年代，剑桥大学才停止从底层出发搭建自己的计算机，并开始从商业贸易商那里购买。这项实际举措直接影响到目前计算机部门的名称：剑桥大学没有计算机科学系，大学里有一个计算机实验室——威尔克斯早期数学实验室的延续。

对于计算机工程实践应用的专注，使得剑桥大学成为孕育新技术的沃土。很多新技术在计算机实验室、工程系或者不同的数学和科学系(即使是我们的数学家也懂得如何进行程序设计)中被提出来，由此吸引了大量跨国公司到此寻找工程技术人才。围绕剑桥大学成长起来的公司网(有时被冠以剑桥产业集群、剑桥现象或硅藻等不同的称谓)代表了美国硅谷之外的少量真正的技术集群之一。那台告诉我应当成为一名芯片设计师的BBC微型计算机就是剑桥生产的，包括其长期的竞争对手——辛克莱频谱。你的手机(以及Raspberry Pi)就包含几个由立足于剑桥的ARM芯片公司设计的处理器。EDSAC问世70年之后，在英国剑桥依然是高科技的家园。

言归正传

在我误打误撞所接受的计算机教育中，最大的缺失是对计算机工作过程的系统认识。在从BASIC开始深入到汇编语言时，我被汇编层面的抽象“困住”了。我可以控制Amiga计算机的硬件寄存器在屏幕上移动小精灵，但对于如何搭建一台自己的计算机则一头雾水。在我花费了另一个十年的时间、取得了几个学位、离开学术界并到博通公司(一家在

剑桥启动并在此寻求工程技术人才的美国半导体公司)工作之后，我才搞清楚这个问题。在这里，我的名片上印着“微电子芯片设计师”(实际上是更漂亮的称谓，“专用集成电路架构师”)的字样。在此期间，我拥有良好的条件与大量该领域的娴熟的从业者一起工作并向他们学习，包括索菲·威尔逊(Sophie Wilson)——原始ARM处理器和BBC微型计算机(和Steve Furber一起)的设计师，还有博通3D图形硬件工程团队的Tim Mamtora，他提供了本书图形处理单元(GPU)章节的内容。

很大程度上讲，撰写本书的目的在于阐述“它是如何工作的”这个话题，正如我18岁时期望的那样。我们尝试覆盖所有现代计算机系统的主要部分，从CPU到易失性随机访问存储器、稳定存储器、网络及接口，以一种中学学生或一年级大学生乐于阅读的方式呈现出来。依托对当前技术发展水平的讨论，我们尝试提供一点历史性知识。绝大多数讨论的话题(尽管不是全部，特别是在技术细节方面)都与1949年维尔克斯的EDSAC工程团队有关系。阅读完本书后，你至少会对计算机的基本工作原理有所了解。我坚信你将发现加强这方面理解的价值，即使你的目标职业是软件工程师并且从未打算设计一台自己的计算机。如果不了解cache的作用，当工作组的大小超出cache或者分配的缓冲区耗尽了cache的关联性时，程序性能的急速下降会令你很惊讶。如果对网络的工作过程一无所知，就很难为数据中心构建一个高效网络。

关于哪些方面本书不会涉及，有必要花点时间阐述一下。对于任何涉及的话题，本书都不是一本综合性技术参考书。关于cache设计、CPU流水线、编译器和网络堆栈，可以编写(有人已经写过)整卷图书。相反，对于每个话题，我们尝试提供一种入门式的讲解，并提供一些深入学习的建议。本书主要专注于通用计算机(本质上讲，PC机)的架构。对于只对特殊目的和应用领域感兴趣的数字信号处理(DSP)和现场可编程阵列(Field Programmable Gate Array，FPGA)等话题，只进行了有限的覆盖。最后，涉及一点定量决策过程的知识，这是良好的计算机架构的核心：如何在访问时间和cache大小之间折中，或者决定是否允许某个子系统一致访问一个属于其他部件的cache？我们无法教你像架构师一样思考。对于高级读者，Hennessy和Patterson的*Computer Architecture: A Quantitative Approach*是这方面不可或缺的参考资料。

曲线上的拐点

首先提出免责声明，此处我愿意分享多年来总结出的几点有益的指导原则。

在计算机架构中，像很多事情一样，有一个收益递减规律。当然，不论是以原始的CPU性能、标准化为能量消耗的CPU性能、存储密度、晶体管尺寸还是媒介上的网络带宽的形式，在任何时候，硬件对所完成的东西有一个限制。但情况往往是，在达到这些理论限制前，工程应用上就会遇到收益递减问题：每一项新增的改进都来之不易，都会引起成本和时间开销的提升。如果将研发成本、系统复杂性 (容易受到漏洞攻击) 或是为提升系统性能而花费的资金绘成图，在某个位置曲线就会急剧弯曲下来。“拐点”的左侧，性能

以一种可预测(甚至是直线！)的方式响应花费支出，在右侧，随着努力的增加性能只是缓慢提升，并逐渐逼近由基础技术限制带来的“墙壁”。

有时候找不到性能的替代品。例如阿波罗登月计划，就是工程学上一个极具吸引力的实例。阿波罗登月计划在“拐点”右侧走出了相当远的距离，从根本上误导了旁观者对于宇航技术成熟度的认识。直到今天——火箭技术、航空电子和材料科学持续发展了50年以后——拐点已经移动了足够远的距离，才允许以合理的成本访问太空，甚至从月球返回。虽然如此，那些以谦卑的态度准确定位拐点位置的团队，为市场带来简单、稳健而又流行的工程系统，然后迅速更新换代，这些团队将最终战胜登月工程。

在对待架构方面，保守和更迭一直铭记于心。迄今为止，我们生产的三代Raspberry Pi芯片都严格地采用相同的系统基础架构、存储器控制器和多媒体，所做的改变只局限于ARM内核的复杂性、少量致命缺陷的修复和时钟频率的提升。这里会出现一些博弈，因为工程师(包括我自己)是富有激情的，他们总希望打破界限。优秀架构师的任务在于精确把握颠覆性改变引发的风险代价，并与预期的利润进行权衡。

迈向基金会

2008年，我们创建了Raspberry Pi基金会，初始目标很简单：应对申请剑桥大学计算机科学专业的学生数量逐步下降的问题。现在，我们看到了令人鼓舞的复苏信号，不论是剑桥还是其他学校，申请数量都超过了20世纪90年代末网络热潮时期的峰值。

我们目睹的一个最显著的变化可能是，新一代年轻人比我们在20世纪80年代时对硬件的兴趣还要高。编写一个汇编语言源程序，在屏幕上移动小精灵不再像以前那么有吸引力，但是在地板上移动机器人却更令人兴奋。我们看到12岁的孩子构建我在20多岁时才引以为豪的控制和传感器项目。我的愿望是，当这些年轻人坐在我孩童时期BBC Micro的新生代面前规划职业生涯时，有一些人可以明白他们将成为优秀的微电子芯片设计师，本书将帮助他们完成这趟旅行。

—Eben Upton，剑桥，2016.05

目　录