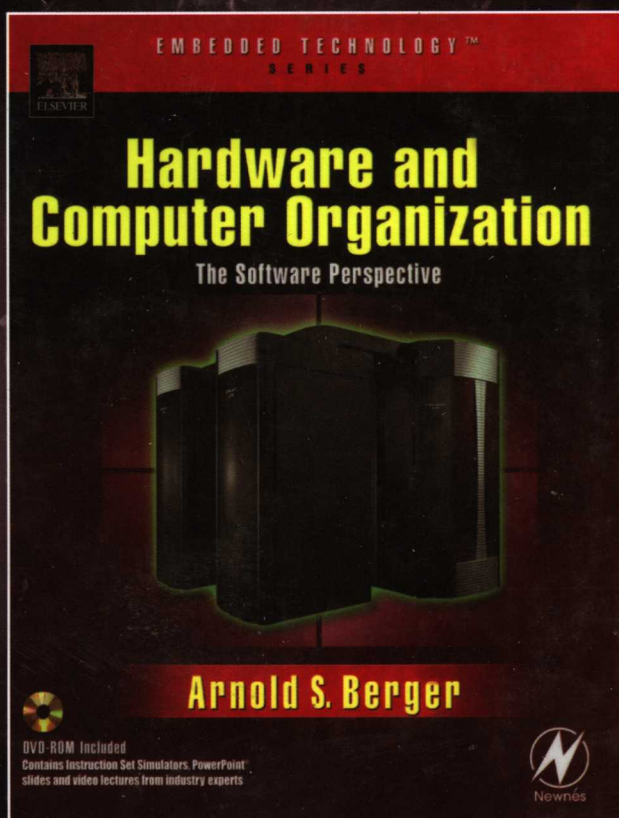




计 算 机 科 学 丛 书

计算机硬件及组成原理

(美) Arnold S. Berger 著 吴为民 喻文健 邓澍军 伍绍贺 译
华盛顿-波泰尔大学 清 华 大 学



Hardware and Computer Organization
The Software Perspective

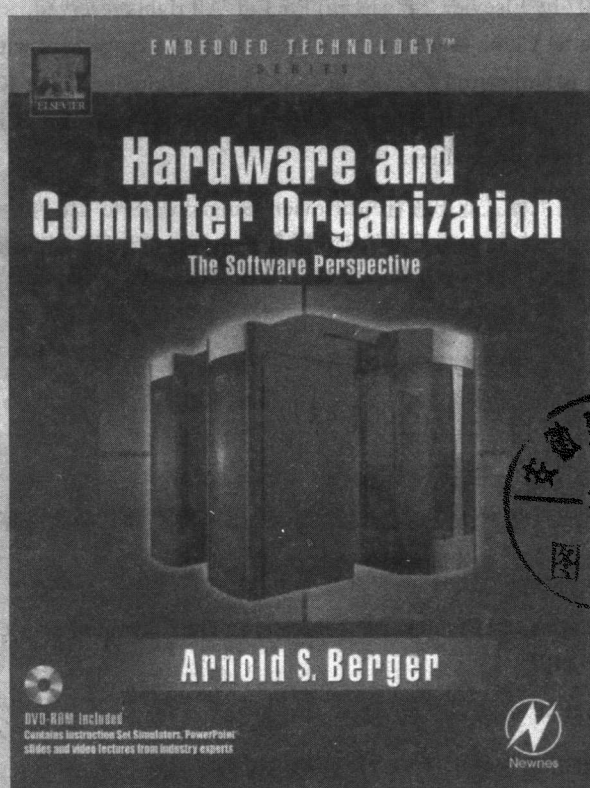


机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

计算机硬件及组成原理

(美) Arnold S. Berger 著 吴为民 喻文健 邓澍军 伍绍贺 译
华盛顿-波泰尔大学 清 华 大 学



Hardware and Computer Organization
The Software Perspective



机械工业出版社
China Machine Press

本书从软件开发者角度出发,详细介绍了现代计算机体系结构,重点讲解如何处理存储器问题以及如何写出能直接与底层硬件交互并充分利用底层硬件的高效汇编代码。

本书主要讲述硬件基础和数字化设计,涵盖现代计算机操作系统下硬件开发的各种元素,从汇编语言讨论软件设计,从宏观角度探讨计算机体系结构,并着重探讨了CISC和RISC两种微处理器体系结构。

本书适合作为高等院校相关专业课程教材,也可供软件开发人员参考。

Arnold S. Berger: Hardware and Computer Organization: The Software Perspective.

ISBN-10: 0750678860

ISBN-13: 978-0750678865

Copyright © 2005 by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

ISBN-10: 981-259-438-8

ISBN-13: 978-981-259-438-9

Copyright © 2007 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 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.

本书简体中文版由机械工业出版社与Elsevier (Singapore) Pte Ltd.在中国大陆境内合作出版。本版仅限在中国境内(不包括中国香港特别行政区及中国台湾地区)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。

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:01-2005-4837

图书在版编目(CIP)数据

计算机硬件及组成原理 / (美)格吉尔(Berger, A. S.)著;吴为民等译. —北京:机械工业出版社, 2007.5

(计算机科学丛书)

书名原文: Hardware and Computer Organization: The Software Perspective

ISBN 978-7-111-21018-4

I. 计… II. ①格… ②吴… III. 硬件 IV. TP303

中国版本图书馆CIP数据核字(2007)第027346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:李东震

北京京北制版厂印刷·新华书店北京发行所发行

2007年5月第1版第1次印刷

184mm×260mm·25.25印张

定价:55.00元(附DVD光盘)

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



机械工业出版社 华章公司

Huazhang Graphics & Information Co., Ltd

教师服务登记表

尊敬的老师:

您好!感谢您购买我们出版的_____教材。

机械工业出版社华章公司本着为服务高等教育的出版原则,为进一步加强与高校教师的联系与沟通,更好地为高校教师服务,特制此表,请您填妥后发回给我们,我们将定期向您寄送华章公司最新的图书出版信息,为您的教材、论著或译著的出版提供可能的帮助。欢迎您对我们的教材和服务提出宝贵的意见,感谢您的大力支持与帮助!

个人资料(请用正楷完整填写)

教师姓名	☐ 先生 ☐ 女士	出生年月	职务	职称: ☐ 教授 ☐ 副教授 ☐ 讲师 ☐ 助教 ☐ 其他
学校	学院		系别	
联系电话	办公: 宅电: 移动:	联系地址 及邮编		
		E-mail		
学历	毕业院校	国外进修及讲学经历		
研究领域				
主讲课程		现用教材名	作者及出版社	共同授课教师
教材满意度				
课程: ☐ 专 ☐ 本 ☐ 研 ☐ MBA 人数: 学期: ☐ 春 ☐ 秋				☐ 满意 ☐ 一般 ☐ 不满意 ☐ 希望更换
课程: ☐ 专 ☐ 本 ☐ 研 ☐ MBA 人数: 学期: ☐ 春 ☐ 秋				☐ 满意 ☐ 一般 ☐ 不满意 ☐ 希望更换
课程: ☐ 专 ☐ 本 ☐ 研 ☐ MBA 人数: 学期: ☐ 春 ☐ 秋				☐ 满意 ☐ 一般 ☐ 不满意 ☐ 希望更换
备注	已出版著作	译著		
	著书	方向一		
	计划	方向二		
	是否愿意从事翻译工作 ☐ 是 ☐ 否 翻译方向			
意见和建议				

填妥后请选择以下任何一种方式将此表返回:(如方便请赐名片)

地址:北京市西城区百万庄南街1号 华章公司营销中心 邮编:100037

电话:(010) 68353079 88378995 传真:(010) 68995260

E-mail: hzedu@hzbook.com marketing@hzbook.com 图书详情可登录<http://www.hzbook.com>网站查询

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

展现在您面前的是这样一本书：它讲的是计算机硬件和体系结构的问题，而且也确实覆盖了这个领域的几乎所有内容，但是，即使是对计算机硬件知识知之甚少的人也能看懂它并从中受益。因为本书是从软件这个视角来介绍计算机体系结构的。

一个软件设计者，如果对其要编程的硬件和体系结构的知识缺乏了解，就失去了很多提高软件性能的手段。然而，要让他们去系统地学习计算机体系结构的知识，再运用这些知识来提高软件性能，则存在一定的困难。因为现今的教科书大都是针对学习体系结构的人而编写的，所以这些教材基本上都是力图完整系统地传授体系结构知识的，并没有考虑一个软件设计者如何利用体系结构的特点设计出高性能的程序。也就是说，对于计算机体系结构，软件开发者在掌握知识和运用知识之间存在着一个鸿沟，本书出现的意义就是填补了这个鸿沟。

本书的讲授风格有别于一般的教科书，它的讲解不是刻板的，而是以一种轻松和诙谐的口吻进行的，就像课堂授课录音的书面记录，读者可以自然地跟进而不会感到疲倦。书中的内容丰富并配有大量的实例，每章后面还有习题并提供了答案（书后和网上）。本书的作者具有丰富的实践经验和教学经验，读者在阅读本书的过程中随处都能体会到。

本书的翻译是4位译者共同努力的结果，其中，吴为民翻译了序言及第1、5、6、7、9、12、15、16章，喻文健翻译了第2、3、4章，邓澍军翻译了第8、10、11章，伍绍贺翻译了第13、14章。最后由吴为民统一校对定稿。对于书中出现的术语，我们基本是按最常用的译法和具体情况进行取舍的。对于原书中出现的错误，我们也完全按照其网站上发布的勘误表进行了改正。尽管如此，疏漏在所难免，我们欢迎和感谢来自读者的任何批评指正。

译者于清华园

2007年1月

前 言

谢谢你购买我的书。我希望你发现本书内容充实且易读，至少，当我着手写这本书时，这是我的目标之一。

本书是我一直在Washington-Bothell大学计算与软件系统系所教授的课程的成果。该课程(CSS 422)，即“硬件和计算机组成”(Hardware and Computer Organization)，是我系本科学士必修的核心课程之一，也是我系课程表中唯一必修的体系结构课程。当我们的学生在学习算法和数据结构、相关的语言、数值方法以及操作系统时，这是他们唯一一次接触到这些知识外壳下真实的原理。由于华盛顿大学是一学年四学期制，所以我就面临着这样一个艰难的挑战：要在10周左右的时间内讲授尽可能多的计算机体系结构的知识。

本书核心的材料是用5年多的时间制成的大约500张Microsoft PowerPoint幻灯片。后来，我将幻灯片中的材料转换成了HTML，这样我就也能通过远程教育(DL)的形式讲授课程了。自从1999年秋首次讲授该课程，我每个学年都要讲上3~4遍。我还通过DL讲授了3次，取得了很好的效果。事实上，DL学生在总体上与在课堂听讲的学生做得一样好，因此，如果你觉得没有时间选修这门课，那么可以用本书来自学这门课程。

本书适合作为从二年级到四年级中在计算机体系结构方面的第一门课。本书相当地独立，所以应该能成为计算机系学生所需学习的唯一硬件课程，掌握了这门课程就能理解他们所编写的代码的涵义。在Washington-Bothell大学(UWB)，这门课程主要是讲授给四年级学生的。作为教员我们发现，通过在其他课程中学习编程概念，达到一定的熟练程度，有助于学生轻松地转向学习低级编程技术。如果本书用于低年级学生，就需要分配额外的时间来熟练掌握汇编语言的编程概念。例如：在介绍某些汇编语言分支和循环结构时，高年级学生很容易掌握这些结构与WHILE、DO-WHILE、FOR及IF-THEN-ELSE结构的相似性，但低年级学生则可能需要更多具体的例子来领会这种相似性。

为什么要写一本关于计算机体系结构的书？在讲授该课程的5年多时间中，我曾4次更换教材。在学期末，当我主持一个非正式的课程听取学生汇报时，他们严厉地批评了我使用过的每本书。几乎每个计算机科学系的学生在体系结构课堂上使用的标准教材都与他们的需要无关。这些学生中的绝大多数在研究生阶段都不会继续学习体系结构，也不会为Intel或AMD设计计算机。因此，他们需要的是理解计算机体系结构及其支撑硬件，以便于编写能在机器上运行的高效的、无缺陷的代码。最近，我确实发现一本教材，至少以我认为应该的方式接近了主题内容，但是我发现该教材仍在几个关键的领域存在不足。从正面看，换成新的教材确实能消除来自学生的抱怨，同时也强化了我的这个观点：并不只我一个人看到了对具有不同视角的教材的需求。遗憾的是，这个教材尽管是一个重大的改进，但仍没有涵盖我认为非常重要的几个领域，因此我下决心写一本教材，本教材确实是以新视角写成的，且没有损失我认为的精髓。

由于UMB校园距华盛顿州Redmond的微软总部不到10英里，所以我们受到了微软文化的强烈影响，这并不奇怪。我的大部分学生只为Windows和Intel体系结构写程序，该体系结构的设计者会让你相信这些计算机是无限快的机器，有无穷的资源。你如何反驳这种观点？

我的学生经常带有挫折感地抱怨，“为什么你要我学这些？”这种情况通常都会发生在期中考试前后。由于我们的校园与波音公司建造737、757飞机的地点华盛顿州Renton和建造宽体767、777飞机的地点华盛顿州Everett大致等距离，用飞机工业做类比通常非常有效，因此我用这个例子简单地回答了他们的问题：“你愿意乘坐一个由某个对飞行原理毫无所知的人设计的飞机吗？”有时这种反问式的回答会很奏效。

本书分为4个主要的主题领域：

1. 硬件和异步逻辑介绍。
2. 同步逻辑、状态机和存储器组织。
3. 现代计算机体系结构和汇编语言编程。
4. 输入/输出、计算机性能、存储器层次，以及计算机组成的未来发展方向。

这些主题领域之间没有明确的界限划分，而且后面章节的主题材料是建立在前面章节的知识基础之上的。然而，我已力图限制这种相互依赖的关系，因此后面的章节可以根据时间和教学提纲的要求进行取舍。

每章末尾都有一些习题，其中奇数号习题的答案位于附录中，偶数号习题的答案则可通过教师资源网址<http://textbooks.elsevier.com/0750678860>得到。

我们在课本中采取了由下而上的方法描述硬件。正如遗传学家用仅包含腺嘌呤、胞嘧啶、鸟嘌呤、胸腺嘧啶（分别简写为A、C、G、T）4种核苷酸的DNA分子就能描述最复杂的有机生命一样，我们用与门（AND）、或门（OR）、非门（NOT）、三态门（TRI-STATE）这4种逻辑构件块就能描述最复杂的计算机或存储系统。严格地说，三态门不是一个类似于与门的逻辑构件块，它更像“粘结剂”，使我们能以某种方式将计算机元件互连，从而避免过高的复杂性。而且，我确实喜欢DNA这个类比，所以我们需要用4个电子构件块与A、C、G、T相类比。

我曾经给一组中学教师做过一个报告，这些教师那时正努力地想在夏季休息期间修一些在职学分。当我为惠普公司的逻辑系统部门工作时，我还是位于科罗拉多Springs的空军院校区的志愿者。这些教师中没有一个人懂计算机，我要用两个小时教给他们这门技术的一些感性知识。我决定从亚里士多德和作为哲学分支的逻辑操作符开始讲解，然后用DNA做类比继续介绍寄存器概念。我似乎正逐渐让他们听懂，但他们却总在打击我的自信心，使我想在他们的点名册上签字解约。无论如何，我认为表明这样的事实是有价值的：即使是最复杂的计算机功能，也能用我们在本书第一部分所介绍的基本逻辑单元来描述。

我们将采用DNA或构件块方法贯穿于本书前半部分中的大部分。我们将从最简单的门开始，构建复合门，再从这些复合门开始，进一步提出和求解异步逻辑方程。我们将学习用布尔代数和卡诺图（Karnaugh Map, K-map）进行真值表设计和化简的方法。习题和实例强调的是将问题用一组规范化描述来陈述，然后转化为一个真值表，再由真值表转化为卡诺图，最后转化为门设计。此时，要鼓励学生使用随书带的DVD光盘中的Digital Works软件模拟器（见下文）在模拟中实际地“构建”电路。我发现这种把抽象设计和实际模拟相结合是一个极好的教学方法。

采用这个方法的好处之一是能使学生们逐渐习惯于在二进制位一级和变量打交道。虽然大多数学生熟悉C/C++的布尔结构，但是一条承载一个变量状态的线对他们来说还是相当新的概念。

介绍了建立任意复杂的异步代数函数的思想之后，我们就加入时钟和同步逻辑的概念。同步逻辑包括触发器、计数器、移位器、寄存器以及状态机。我们在该领域实际付出了很多

努力，在稍后讲到微代码和指令分解时，还要将这些概念重新介绍几次。

本书的中间部分着重介绍计算机系统的体系结构。尤其是，我们会非常关注存储器到CPU的接口。我们将利用在前面章节学到的知识设计简单的存储器系统和译码电路。我们还简要考察了存储器定时，以便于更好地理解系统设计的一些更全局化的问题。

接下来我们将转而介绍68K、ARM以及x86处理器系列的体系结构，这同时也将是对汇编语言编程的介绍。

每种处理器体系结构都单独讲解，以使读者可跳过某种结构的介绍而不会产生过多的不连贯性。

本书确实在三个体系结构中都强调了汇编语言编程，其原因有两个方面：首先，汇编语言不一定是作为课程表的一部分为计算机科学系学生讲授的，故这可能是他们在机器级接触编程的唯一机会。即使你作为计算机系的学生可能从不需要编写汇编语言程序，你也很可能要在汇编语言级来调试C++程序的某些部分，因此，这是一个很好的学习机会。而且，通过考察这三个截然不同的指令集，我们将实际强化这种观念：一旦你理解了一种处理器的体系结构，你就能用汇编语言对其进行编程，这就引出了学习汇编语言的第二个原因。汇编语言是从软件开发者的角度学习计算机体系结构的一个很好的出发点。

我是“赛博士”(Dr. Science)的一个狂热爱好者，他经常出现在国家无线广播电台中，并在大学校园中巡回演讲，他的著名时髦语是：“我在某某学科具有硕士学位。”不过，我在赛博士的讲座上听到这样一句话：“我喜欢审视一列列的随机数字，并从中发现模式。”我记住了这句话，并经常用在我的课堂上，以描述如何能够通过看起来很随机的机器语言指令集来逐渐领会计算机体系结构。我只能想像一群摩托罗拉的CPU设计师和工程师围坐在餐馆的桌旁，桌子上的比萨饼盘子散布在各处，他们正试图为最后几条指令计算出正确的位模式，以避免得到一个膨胀的、无效率的微代码ROM表。如果你是一个学生，并且读到这里还没有任何感觉，那么也不要着急。

本书最后一部分又返回来重新考察了计算机体系结构的一般问题。我们将考察CISC与RISC，以及诸如流水线、高速缓存、虚拟存储器和存储器管理等这些现代技术。然而，最重要的主题还是计算机性能，我们将不断地返回到有关软件到硬件接口、编码方法和硬件之间的相互影响等问题上来。

本书的一个独特之处在于随书所附的DVD光盘中的材料，其中包含了下列程序，用来与本书配合使用：

- Digital Works (免费软件)：一个硬件设计和模拟工具。
- Easy68K：一个汇编器/模拟器/调试器免费软件包，用于摩托罗拉（现在免费级别的）68 000体系结构。
- x86emul：一个汇编器/模拟器/调试器共享软件包，用于x86体系结构。
- GNU ARM工具：ARM开发者工具箱，带有来自自由软件基金会（Free Software Foundation）的指令集模拟器。
- 来自11位业界专家的关于硬件设计和开发方面重要主题的视频讲座。

ARM公司有一个极好的工具套件，你可直接从ARM获得。它带有一个免费45天的评估许可证，这对于我们的课程来讲应该足够用了。遗憾的是，我未能通过谈判与ARM公司达成一个许可证协议，使我能将这些ARM工具包含在随书的DVD光盘中。该工具套件极为优秀和易用。如果你想花费一些额外的时间来考察世界上最流行的RISC体系结构，那么就直接和ARM

公司联系，友好地寻求ARM套件的拷贝吧。告诉他们是我让你这样做的。

我还在CSS 422课上广泛地采用了Easy68K汇编器/模拟器。它性能良好，并附有很多调试功能。而且，由于它是免费软件，所以无需考虑许可证和评估期限这些后顾之忧。然而，我们还要对本书中的其他工具做一些引用，因此，最好在你打算使用时再安装它们，而不是在课程开始时就安装它们，这也许是一个好主意。

DVD光盘包括的11个视频讲座，内容与本书各种主题相关，演讲者来自于计算机体系结构领域的专家。这些视频文件是在2004年UWB的沃辛顿技术捐赠基金提供的资助下制作的。每个讲座都是15到30分钟的技术讲解。我希望你花时间看一下，并将它们融汇到相关主题的学习过程中。

虽然编辑、我的学生们以及我本人都已将本书读过几遍了，但据墨菲法则的预言，本书还有存在错误的巨大可能性，因为毕竟它是软件。因此，如果你在本书中看到错误，那么请告诉我，并将你的意见发送到：aberger@u.washington.edu。我将保证这些修正意见能在我的华盛顿大学的网站（<http://faculty.uwb.edu/aberger>）上张贴出来。

最后一点我想说的是，课本只能做到这样。无论你是正在读本书的学生还是老师，请尽量寻找专家和原始资源。James Patterson教授在2004年7月期的《Physics Today》（今日物理）中写到：

当我们想了解某件事物时，总有一种想在课本中快速寻求答案的倾向。这常常是奏效的，但我们需要养成查看原始论文的习惯。因为课本上的内容通常是把事实经过二次或三次简化而来……

让我们开始吧。

Arnold S. Berger
Sammamish, 华盛顿

致 谢

首先，我要感谢Washington-Bothell大学计算与软件系统系的前主任William Erdly教授的赞助和支持。Erdly教授首先在1999年秋聘用我为助教授，并要求我讲授一门称为“硬件和计算机组成”的课，虽然我那时想讲的是关于嵌入式系统设计的课程。

然后Erdly教授就为我提供经费支持，将我的“硬件和计算机组成”课程中一系列PowerPoint幻灯片讲义转换成一系列HTML形式，供在线课程使用。这些内容成了本书的核心材料。

在Charles Jackels教授和Frank Cioch教授任执行主任期间，都在完善在线材料和将多媒体引入远程教育实践方面给予了我支持。他们的支持使我认识到了课堂中技术的教学价值。

我还感谢沃辛顿基金会的Richard P.和Lois M.，他们提供的2004技术资助奖使我能在全美旅行并做计算机体系结构方面的简短视频录像。我也想感谢那11位演讲者能抽出时间参与该计划。

Newnes图书出版公司的选题编辑Carol Lewis认识到了我的方法的价值，我为此感谢她。遗憾的是，在这本书最终完成之前，她就离开了Newnes。Elsevier出版社的Tiffany Gasbarrini和Borrego出版社的Kelly Johnson使本书最终得以面世，谢谢你们两位。

本书很大程度上是那些曾经学习CSS 422课程的学生所设计的。他们的期末评估和反馈对于帮助我认识到如何构思本书具有无法估量的价值。

最后，也是最重要的，我要感谢我的妻子Vivian。没有她的支持和理解，我不可能写出这样篇幅的一本书。

目 录

出版者的话	
专家指导委员会	
译者序	
前言	
致谢	
第1章 硬件体系结构简介	1
1.1 引言	1
1.2 计算技术简史	1
1.3 数制	9
1.4 将十进制数转换为各种基数的数	20
1.5 工程符号	21
总结	22
参考文献	22
习题	22
第2章 数字逻辑简介	24
2.1 引言	24
2.2 电子门描述	32
2.3 真值表	36
总结	38
参考文献	38
习题	38
第3章 异步逻辑简介	40
3.1 引言	40
3.2 布尔代数定律	41
3.3 卡诺图	45
3.4 时钟和脉冲	50
总结	55
参考文献	55
习题	55
第4章 同步逻辑简介	58
4.1 引言	58
4.2 触发器	59
4.3 存储寄存器	68
总结	74
参考文献	75
习题	75
第5章 状态机简介	79
5.1 引言	79
5.2 现代硬件设计方法	96
总结	98
参考文献	98
习题	99
第6章 总线组织和存储器设计	103
6.1 总线组织	103
6.2 地址空间	115
6.3 直接存储器访问	128
总结	129
参考文献	130
习题	130
第7章 存储器组织和汇编语言编程	134
7.1 引言	134
7.2 标号	143
7.3 有效地址	147
7.4 伪操作代码	154
7.5 数据存储伪指令	155
7.6 汇编语言程序的分析	156
总结	158
参考文献	158
习题	158
第8章 汇编语言程序设计	162
8.1 引言	162
8.2 汇编语言和C++	175
8.3 堆栈和子程序	180
总结	186
参考文献	186
习题	186
第9章 高级汇编语言编程	192
9.1 引言	192
9.2 高级寻址模式	192
9.3 68000指令	194
9.4 移动指令	195
9.5 逻辑指令	195
9.6 其他逻辑指令	196

9.7 68000指令总结	199	12.1 引言	266
9.8 用TRAP#15指令模拟I/O	201	12.2 中断	267
9.9 编译器和汇编器	203	12.3 异常	270
总结	216	12.4 Motorola 68K的中断	270
参考文献	216	12.5 模数(A/D)转换和数模(D/A)转换	274
习题	216	12.6 A/D和D/A转换器的分辨率	286
第10章 Intel x86体系结构	220	总结	288
10.1 引言	220	参考文献	288
10.2 8086 CPU的体系结构	221	习题	288
10.3 数据寄存器、变址寄存器和指针寄存器	223	第13章 现代计算机体系结构简介	292
10.4 标志寄存器	226	13.1 处理器体系结构, CISC、RISC及DSP	293
10.5 段寄存器	226	13.2 流水线简介	296
10.6 指令指针 (IP)	226	总结	305
10.7 存储器寻址模式	228	参考文献	305
10.8 x86指令格式	231	习题	306
10.9 8086指令集总结	233	第14章 存储器、高速缓存和虚拟存储器	308
10.10 数据传送指令	234	14.1 高速缓存简介	308
10.11 算术指令	235	14.2 虚拟存储器	321
10.12 逻辑指令	235	14.3 页	323
10.13 字符串操作	236	14.4 转换旁路缓冲器 (TLB)	324
10.14 控制转移	237	14.5 保护	325
10.15 8086体系结构的汇编语言程序设计	239	总结	326
10.16 系统向量	241	参考文献	327
10.17 系统启动	241	习题	327
总结	241	第15章 计算机体系结构的性能问题	329
参考文献	242	15.1 引言	329
习题	242	15.2 硬件和性能	329
第11章 ARM体系结构	244	15.3 最佳习惯	342
11.1 引言	244	总结	343
11.2 ARM体系结构简介	245	参考文献	344
11.3 条件执行	249	习题	344
11.4 桶式移位器	250	第16章 未来发展趋势与可重构硬件	346
11.5 操作数大小	250	16.1 引言	346
11.6 寻址模式	251	16.2 可重构硬件	346
11.7 堆栈操作	253	16.3 分子计算	354
11.8 ARM指令集	255	16.4 局部时钟	355
11.9 ARM系统向量	263	总结	358
总结	264	参考文献	358
参考文献	264	习题	358
习题	265	附录 奇数号习题答案	360
第12章 与外部接口	266	索引	382

第1章 硬件体系结构简介

学习目标

- 描述计算器件的演化以及大多数计算机器件的组织方式；
- 在二进制、八进制、十六进制数之间做简单的转换，并解释这些数制对计算器件的重要性；
- 表明计算机硬件的原子元件和逻辑门的使用方式，并详述支配它们操作的规则。

1.1 引言

今天，当一排计算机围绕在我们周围帮助我们管理日常生活时，我们通常会视这些为理所当然。对于正在学习计算机体系结构和数字硬件的你来说无疑是很好的理解并习以为常，而且你可能在个人计算机和工作站上已编写了数不清的程序，计算机技术已经进展到如此程度，即每个任天堂GameBoy游戏机的计算能力都百倍于用于首次执行水星太空任务的计算机系统。

1.2 计算技术简史

计算机自出现已经历了漫长的几百年的时光。中国算盘、带有传动装置和齿轮的计算器以及第一台模拟计算机都是计算机器的例子。我们将要介绍的计算机器产生于20世纪40年代，因为第二次世界大战中的炮兵需要一种更精确的方法来计算从战舰上发射炮弹的轨迹。

今天，计算机变得如此普遍流行的主要原因是集成电路制造技术的进步。在加利福尼亚的San Jose北部和Palo Alto南部，曾经主要以橘林闻名的地方如今已被称为硅谷。硅谷是很多公司的大本营，而这些公司正是集成电路技术的推动者。Intel、AMD、Cypress、Cirrus Logic等等这些名字都在全世界家喻户晓。

大约30年前，Intel的创始人之一Gordon Moore观察到，安置于单个硅芯片上的晶体管密度每18个月翻一番。该论点自从Moore首次指出以来一直相当准确，并被公认为摩尔定律。与其他指标相比，存储器容量更适于用作说明摩尔定律准确性的例子。图1-1是存储器容量对于时间的半对数图。很多电路设计者和器件物理学家还在争论摩尔定律持续下去的可能性。晶体管不能无限地缩小，要生产如此小尺寸的硅圆片，制造者也负担不起制造设备的成本。在某个点上，量子物理学定律将开始深刻地改变这些微小晶体管的特性。

今天，我们能够将数以亿计的晶体管（即我们用来建造逻辑门的“活动”开关器件）安置在单个硅片上，边长大概为2cm。从设计计算机芯片的角度看，这个重大突破发生在Mead和Conway¹描述了一种通过编写软件产生硬件设计的方法之后。这个称为硅编译（silicon compilation）的方法导致了硬件描述语言（hardware description language, HDL）的产生。硬件描述语言如Verilog和VHDL能够使硬件设计者编写出与C程序设计语言极其相似的程序，然后将该程序编译成一个处方（recipe），半导体制造者就用该处方制造芯片。

回到开始。第一代计算机的引擎由机械装置组成。算盘、加法机、纺织机穿孔卡读卡器属于这一类。第二代跨越了1940~1960年这段时期。这个时期用电子器件（即真空管）作为活动

器件或开关元件。即使一个微型的真空管也比硅圆片上的一个晶体管大几百万倍，其消耗的功率是晶体管的几百万倍，而其使用寿命则比晶体管小几百倍或几千倍。虽然真空管计算机比前一代的机械计算机快得多，但仍比现在的计算机慢几千倍。如果你是20世纪50年代B级科幻小说改编电影的爱好者，这些计算机就是那些充满房间，身上灯光闪烁，仪表针跳动的东西。

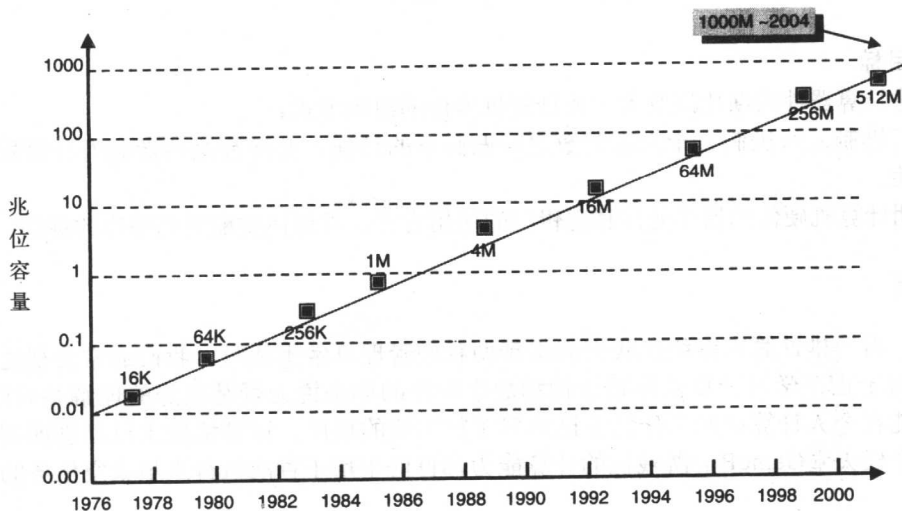


图1-1 动态随机存储器 (DRAM) 的容量随时间的增长。注意其半对数行为，表明了摩尔定律

第三代大致是从1960年到1968年这段时期。这个时期，晶体管取代了真空管，计算机突然开始有能力做实际工作了。诸如IBM、Burroughs、Univac这样的公司都建造了大型机，IBM 360系列就是当时大型机的典型例子。当时，Xerox的研究者也在他们的Palo Alto研究中心 (Xerox PARC) 进行一些关于人机接口方面的开创性工作。在那里，他们的研究成果后来演变成为计算机网络、Windows操作系统以及广泛使用的鼠标。程序员们不再用机器语言和汇编语言编程，而是开始使用FORTRAN、COBOL和BASIC。

第四代大致是从1969年到1977年，是小型计算机的时代。小型计算机是大众化的计算机，它虽然还不完全是个人计算机，但它已将计算机搬出了由穿白外套的技术人员维护的计算机房的消毒环境，搬进了我们的实验室。小型计算机还象征着集成电路（即单一封装的逻辑函数集块）的出现，它替代了诸如晶体管、电阻这样布置于印刷电路板上的单个电子零件（称为分立器件）。这段时期出现了小规模和中规模的集成电路。诸如数据设备公司 (DEC)、Data General以及HP这样的公司都建造了这一代的小型计算机²。在这段时期中，还提出了简单的集成电路微处理器，并由Intel、Texas Instruments、Motorola、MOS Technology以及Zilog等公司将其商品化。早期最能代表这一代的微计算机器件是Intel的4004、8008和8080，Texas Instruments的9900，以及Motorola的6800。第四代的计算机语言是：汇编、C、Pascal、Modula、Smalltalk和Microsoft BASIC。

我们目前处于第五代，虽然有争论认为第五代结束于Intel 80486微处理器，奔腾 (Pentium) 代表第六代，但我们将忽视这个不同意见，直至它被更广泛地接受。半导体制造技术的进步是对第五代计算机特色的最佳刻画，当今半导体工艺代表了称为超大规模集成电路 (Very Large Scale Integration, VLSI) 的技术。下一步，甚大规模集成电路 (Ultra Large Scale Integration, ULSI) 既非就在眼前，也不是即将来临。AMD研究员 (Fellow) Daniel Mann博士³最近告诉我，现代AMD Athlon XP处理器包含将近6千万个晶体管。

第五代还见证了个人计算机和操作系统的成长，后者还是该种机器的焦点。由标准操作系统控制的标准硬件平台使数以千计的开发者为这些系统编写程序。就软件而论，占支配地位的语言变为ADA、C++、Java、HTML和XML。此外，基于通用建模语言（UML）的图形设计语言开始出现。

3

对当前计算机的两种观点

现代计算机已变得更快、更强大，但是在很多年里计算机的基本体系结构本质上未变。现今我们可以对这种机器持两种等价的观点：硬件观点和软件观点。毫不奇怪，硬件观点关注的是机器，并确实考虑到软件与其存在的这种前提相关。从5万英尺远，我们的计算机看起来如同图1-2所示。

在本课程中，我们将主要关注CPU和存储器系统，并少许考虑驱动硬件的软件。我们将略微谈及I/O，因为I/O对于好的计算机来说是不可缺少的。

软件开发者的观点大致等价，但其视角却有所变化。图1-3从软件开发者的视角描述了计算机。注意图中显示的对系统的观察有点问题，因为用户接口与应用程序的直接通信并不总是清楚的，在很多情况下，用户接口首先与操作系统通信。然而，我们可以用稍微宽松的眼光来看待该图，将其看成信息流，而不是控制流。

抽象级别

关于计算机设计，一个更现代的概念是抽象级别的思想。每个级别提供的是它下面级别的抽象，位于最低级别的就是硬件。为了控制硬件，就有必要生成一些小的，称为驱动程序（driver）的程序，用来实际操纵硬件的各个控制位。

位于驱动程序上面的是操作系统和其他系统程序。操作系统（OS）通过一组标准的应用编程接口（Application Programming Interface, API）与驱动程序通信。API提供了一种结构，通过它，抽象级别中的上一级别可与下一级别通信。这样，为了从计算机的键盘读一个字符，就要有一个低级别的驱动程序，当一个按键被敲击时，激活该驱动程序。操作系统通过其API与该驱动程序通信。

在接下来的级别中，应用软件通过系统API与操作系统通信，这又是一个对较低级别的抽象，

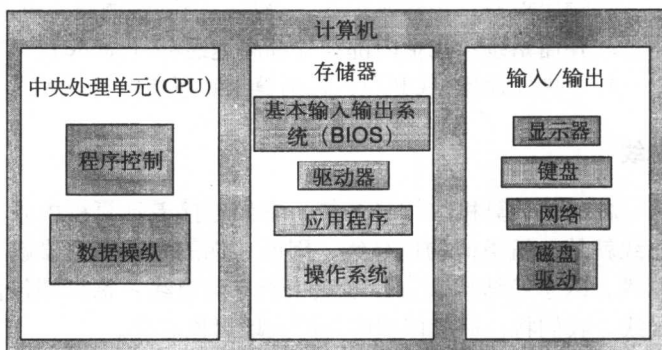


图1-2 计算机的抽象观察。三个主要的要素是：控制和数据处理器、输入和输出（即I/O器件）以及运行于机器上的程序

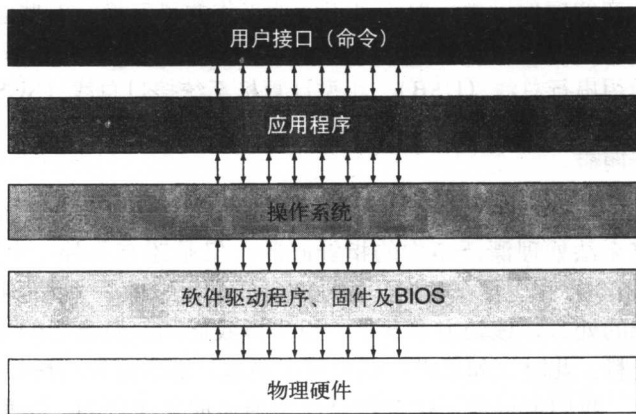


图1-3 根据系统的抽象级别表示计算机

4

使得硬件和驱动程序在行为上的个体差异可以被忽略。然而，我们需要稍加注意，不要对图1-3的观点理解得过于字面化。我们还可以争辩说，应用程序和操作系统级别应该颠倒过来，因为用户也通过操作系统级别与应用程序进行交互。这样，鼠标和键盘输入实际上是通过操作系统传送给应用程序的，而不是直接从用户传送给应用程序的。无论如何，你现在了解了这些概念。

可以将台式PC机为代表的计算机硬件看作是由4个基本部件组成的：

1. 输入设备：包括鼠标、键盘、麦克风、磁盘、调制解调器及网络等组件。
2. 输出设备：包括显示器、磁盘、调制解调器、声卡和喇叭以及网络等组件。
3. 存储系统：包括内部和外部高速缓存（cache）、主存储器、视频存储器及磁盘。
4. 中央处理器（CPU）：包括算术和逻辑单元（ALU）、控制系统及总线。

总线

总线是计算机的神经系统，它们连接着计算机内外的各种功能块。在计算机内部，一个总线就是一组类似的信号线，因而，你的奔腾处理器就有一个32位地址总线和一个32位数据总线。以总线结构来看，这意味着有两束线，每束都包含着用于完成共同功能的32个单独信号线。我们将在后续课程中更深入地讨论总线。

一台典型的计算机有3个总线：一个用于存储器地址，一个用于数据，一个用于状态（管理和控制）。产业界的标准总线有PCI、ISA、AGP、PC-105、VXI，等等。由于为这些产业标准总线所做的信号定义和时序要求是由维护这些标准的团体谨慎地控制的，所以来自不同制造商的硬件设备一般均可正确地工作并可互换。有些总线相当简单，只有一条线，但传向该线的信号却相当复杂，以致需要特殊的硬件和标准协议来理解信号。这种类型总线的例子有：通用串行总线（USB）、小型计算机系统接口总线（SCSI）、Ethernet以及Firewire。

5

存储器

从软件开发者的观点看，存储系统是计算机最显而易见的部件。如果没有存储器，我们将无法处理漂浮不定的指针问题。但是还有另外一个方面：计算机存储器是存储程序代码（指令）和变量（数据）的地方。我们可以做一个关于指令和数据的简单类比。考虑一个烤蛋糕的处方，该处方本身是一组告诉我们如何做蛋糕的指令，数据代表指令要对其施以操作的配料。此时，如果没有面粉可供筛选，那么筛选面粉的指令就是没有意义的。

我们也可基于速度将存储器描述成层次化的。这里速度的意思是当计算机需要时，能以多快的速度从存储器中检索到数据。最快的存储器也是最昂贵的，因此，当存储器访问时间变慢时，每个位的成本就会降低，我们就能得到更多的存储器。最快的存储器也是最靠近CPU的存储器。因此，CPU就可能会有少量的片上数据寄存器或存储地址，几千的片外cache存储器地址，几百万的主存储器地址，以及几十亿的磁盘存储器地址。最快的片上存储器的访问时间大约是最慢的硬盘存储器的访问时间的一万分之一。两种存储器的成本比率有点难计算，因为最快的半导体存储器是片上cache存储器，而你不能在微处理器本身以外单独购买它。然而，如果我们要估计PC机中每10亿字节主存储器的成本与每10亿字节硬盘存储器的成本（计入邮寄折扣）的比率，则会发现在成本上，具有20~40ns平均访问时间的最快的半导体存储器比具有1ms平均访问时间的硬盘存储器多300倍。

今天，由于PC产业的规模经济，存储器便宜得难以置信。一个具有512MB存储地址容量的标准存储器模块（SIMM）成本大约为60美元。一种称为动态随机存储器（dynamic random access memory, DRAM）的存储技术占据着PC存储器的统治地位。DRAM有若干种变型，我