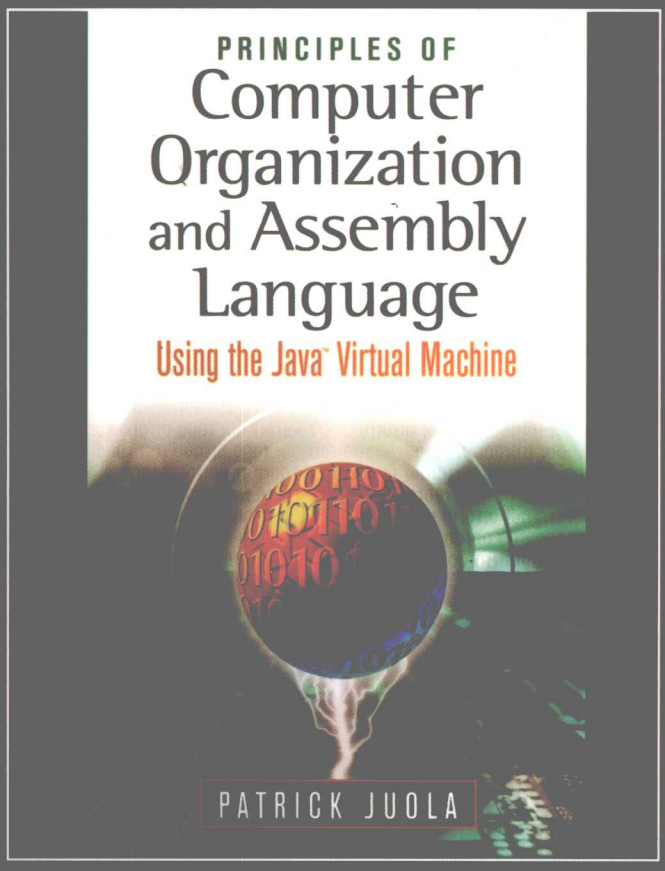
HZ BOOKS
华章教育

计 算 机 科 学 丛 书

计算机组成及汇编语言原理

(美) Patrick Juola 迪尤肯大学 著 吴为民 艾丽华 张大伟 译



Principles of Computer Organization and Assembly Language
Using the Java Virtual Machine

机械工业出版社
China Machine Press
PEARSON Education

计 算 机 科 学 丛 书

计算机组成及汇编语言原理

(美) Patrick Juola 迪尤肯大学 著 吴为民 艾丽华 张大伟 译

Principles of Computer Organization and Assembly Language Using the Java Virtual Machine

机械工业出版社
China Machine Press

本书以Java虚拟机为基础介绍计算机组织和系统结构。前半部分涵盖了计算机组织的一般原理，以及汇编语言编程的艺术，后半部分关注于各种不同CPU在系统结构上的特殊细节，包括奔腾、8088、Power系统结构以及作为典型嵌入式系统控制芯片例子的Atmel AVR。

本书全面反映了IEEE和ACM所推荐的标准计算机体系结构及组成课程应涵盖的知识要点，适用范围广，可作为高等院校计算机及相关专业计算机组成课程的教材。

本书版权登记号：图字：01-2007-1831

图书在版编目（CIP）数据

计算机组成及汇编语言原理 /（美）卓拉（Juola, P.）著；吴为民等译. —北京：机械工业出版社，2009.9

（计算机科学丛书）

书名原文：Principles of Computer Organization and Assembly Language: Using the Java Virtual Machine

ISBN 978-7-111-27785-9

Ⅰ.计… Ⅱ.① 卓… ② 吴… Ⅲ.① 计算机体系结构 ② 汇编语言 Ⅳ.TP303 TP313

中国版本图书馆CIP数据核字（2009）第123710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：刘立卿

北京市荣盛彩色印刷有限公司印刷

2010年1月第1版第1次印刷

184mm × 260mm · 15.75印张

标准书号：ISBN 978-7-111-27785-9

定价：39.00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：hzjsj@hzbook.com

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分社较早意识到“出版要为教育服务”。自1998年开始，华章分社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。华章分社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

译者序

当前，对于计算机组成与系统结构类的本科课程，在教学上的主要困难之一是难以选择一个合适的教学用体系结构。能清楚体现计算机组成和体系结构原理的芯片早已过时；而对于先进的奔腾机，这些基本原理则淹没于复杂的实现方法和策略中。

本书作者意识到了目前计算机组织和系统结构在教学选材上的困难，并采取JVM作为教学体系结构。这是从新的角度进行的有益尝试。JVM非常简单、易于理解，因而可能会成为系统结构教学的最佳用机之一。但JVM毕竟与真实计算机存在物理差别，为表明这种差别，作者也有针对性地介绍了其他几种典型的体系结构。

本书的特点是内容广泛且有一定深度，从最基本的电子器件、二进制表示和计算，到jasmin汇编语言程序设计，再到现实世界中存在的计算机系统结构，最后到JVM高级编程课题，几乎涵盖了所有相关的主题。并且，在每个章节都提供了习题，以巩固知识。

本书适合于作为大学二、三年级相关课程的教材或教学参考书。学生们通过一学期的学习，就能基本掌握计算机组成的基本原理及汇编语言编程。当然，如果学生们已经掌握了计算机的最基础知识，再学习本书则效果更好。

本书由三位老师合作翻译。吴为民翻译了第1、2、3、4、10章以及附录A、C、D、E，艾丽华翻译了第5、6、7、8、9章，张大伟翻译了附录B。由于本书的翻译工作是在繁忙的教学、科研工作之余完成的，难免有疏漏之处，欢迎各位读者给予批评指正。

译者

2009年10月

前 言

本书内容

这是一本关于Java虚拟机（Java Virtual Machine，JVM）组织和系统结构的书。JVM是处于Java语言核心的软件，并出现在大多数计算机、Web浏览器、PDA以及网络化附属设备中。本书还涵盖了计算机组织和系统结构的一般原理，并以其他流行（或不那么流行）的计算机为例加以说明。

这不是一本关于编程语言Java的书，虽然具备Java语言或类Java语言（C、C++、Pascal、Algol等）的一些知识会有所帮助。本书是一本关于Java语言如何使事件发生以及计算如何产生的书。

这本书的写作开始于一个现代技术的实验。当我开始任教于目前的大学时（1998年），计算机组织和系统结构课程用的主要是运行MS-DOS的8088，这个编程环境实质上与修这门课的二年级学生年龄相当。（遗憾的是，这种时间上的迟滞相当普遍。当我在本科修同样的课程时，所学系统结构的相应计算机只比我“年轻”2年。）根本问题是现代奔腾4芯片不是特别好的教学用系统结构。它加入了有20年历史的8088的所有功能，包括其局限，并提供了复杂的变通方法。由于这个复杂性问题，就难以在不详细引用早已过时的芯片集的情况下解释清楚奔腾4的工作原理。教科书主要讲解的是较简单的8088，然后作为扩展和后续思考来描述实际要使用的计算机。这就好比在福特A型上学习汽车力学，后来只讨论如催化式排气净化器、自动驾驶、基于钥匙的点火系统等重要概念。计算机系统结构课程不应被迫成为计算历史的课程。

与此不同的是，我想采用一种易于理解的系统结构来教这门课，该系统结构结合了现代原理且本身对学生有用。由于每个运行Web浏览器的计算机都结合了JVM的一个副本作为软件，因此几乎每个当今的计算机都已经有了兼容的JVM供其使用。

因而这本书涵盖了计算机组织和系统结构的核心方面：数字逻辑和系统、数据表示以及计算机组织/系统结构。本书还描述了一种特定系统结构JVM的汇编级语言，并且介绍了其他常见的系统结构（如英特尔奔腾4和Power PC）作为支持例子但不作为重点。正如IEEE计算机学会和美国计算机协会所推荐的，本书尤其适合作为计算机系统结构和组织的标准二年级课程。[⊖]

组织

本书包含两个部分。前半部分（第1～5章）涵盖了计算机组织和系统结构的一般原理，以及汇编语言编程的艺术/科学，并采用了JVM作为例子来阐明这些原理如何起作用（在数字计算机中如何表示数？加载器做哪些事情？格式转换涉及哪些事情？），以及JVM汇编语言编程中一些必要的细节，包括对操作代码的详细讨论（操作代码i2c要做哪些事情，它是如何改变堆栈的？运行汇编器的命令是什么）。本书的后半部分（第6～10章）关注于各种不同CPU在系统结构上的特殊细节，包括奔腾、它的老亲戚8088、Power系统结构，以及作为典型嵌入式系统控制芯片例子的Atmel AVR。

⊖ “Computing Curricula 2001,” 2001年12月15日，最终草案；特别参见关于课程CS220的推荐意见。

读者

这个框架将使得本书被广大读者和众多课程所使用，这是我的希望和信念。本书应能成功地服务于以软件为中心的计算机产业。对于那些主要感兴趣于将编程语言作为基础来学习抽象的计算机科学的人来说，JVM对计算的基本操作提供了一个简单、易于理解的介绍。作为编译器理论、编程语言或操作系统课程的基础，JVM是一个便利和可移植的平台和目标系统结构，比任何单芯片或操作系统有更广的可用性。作为进一步学习（特定平台的）各种计算机的基础，JVM提供了一个有用的解释性教学系统结构，该系统结构不仅可向目前的奔腾，而且可向在未来可能取代或支持奔腾的其他系统结构，实现平滑的、有原则的过渡。对于有兴趣学习计算机如何工作的学生来说，本书将提供有关大量不同平台的信息，以增强使用实际计算机和系统结构的能力。

如上所述，本书主要是作为本科二年级的单学期课程的教科书。前四章给出了理解计算机组织、系统结构以及汇编语言编程所需的核心材料。假设读者已经有了高级命令性语言的一些知识，并且熟悉高中代数（不是微积分）。在此基础上，教授（和学生）在选择主题方面有某种程度的灵活性，这取决于环境和具体问题。对于Intel/Windows工作组，关于8088和奔腾的章节就是有用和相关的，而对于有老式苹果机或基于Motorola微处理器实验室的学校，关于Power系统结构的章节更为相关。讲述Atmel AVR的一章可为嵌入式系统或微计算机实验室工作奠定基础，而高级的JVM课题将是打算以JVM系统结构为基础实现基于JVM的系统或编写系统软件（编译器、解释器等等）的学生之兴趣所在。进度快的课程甚至可能会涵盖本书所有的主题。书中还提供了附录供参考，因为我们相信，好的教科书应该在课程结束后仍是有用的。

致谢

没有Duquesne大学的学生，尤其是在计算机组织和汇编语言课程中参加我的实验的学生们，就不会有这本书。还要感谢我所在的系、学院以及大学所提供的帮助，尤其是来自Philip H.和Betty L. Wimmer家庭基金会的基金支持。我还要感谢我的读者，尤其是Pittsburgh大学的Erik Lindsley对早期草稿的宝贵意见。

没有出版商，本书永远不会与读者见面。因此我还要感谢Tracey Dunkelberger和Kate Hargett两位编辑，并通过他们向Prentice Hall出版集团致谢。我要向所有的审阅人致以谢意：Western Illinois大学的Mike Litman、Texas Tech大学的Noe Lopez Benitez、Arkansas Tech大学的Larry Morell、加州州立大学（Channel Islands）的Peter Smith、路易斯安娜州立大学（Shreveport）的John Sigle、以及密苏里大学（Columbia）的Harry Tyrer。同样，没有软件也不会有这本书。除了显然要感谢Sun公司发明Java的那些人以外，我特别地想要感谢jasmin的作者Jon Meyer，感谢他编写的软件以及他提供的有益支持。

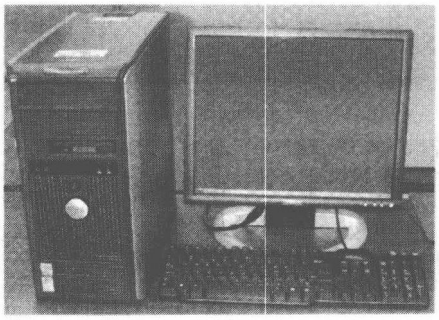
最后，我还要感谢我的妻子Jodi，她为大多数示意图绘制了最初草图。更重要的是，在本书的长期写作过程中，她一直在努力容忍我，并且仍然愿意与我生活在一起。

目 录

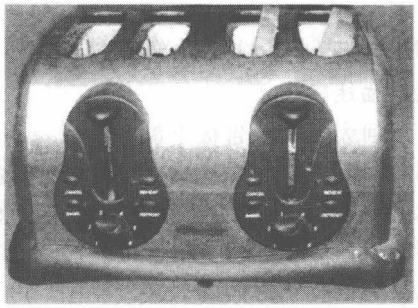
第一部分　假想计算机

第1章　计算和表示

1.1　计算



一台计算机



也是一台计算机

1.1.1　电子设备

有多少人真正知道计算机是什么？如果你问这个问题，大多数人会指向某人桌子上（或者也许是某人公文包中）的一组盒子——这可能是一组由灰色塑料包装的、外形呆板的方形盒子，并且纠缠了一堆连线，类似于一台电视机。如果穷追细节，他们会指向某个盒子，称之为“计算机”。不过当然也有计算机是隐藏在各种日常电子部件内部的，它们的作用可能是确保汽车的燃油效率足够高，解释来自DVD播放机的信号，甚至是确保早餐面包烤得恰到好处。但是对于大多数人来说，计算机仍然是你从电子商店购买的盒子，并且还要常常比较其存储量（如位数和字节数）和频率（如千兆赫），但很少有人真正明白其含义。

用功能的术语来说，计算机就是一台高速的计算器，平均每秒能执行几千、几百万，甚至几十亿的简单算术操作，这些操作由存储的程序所规定。大概每千分之一秒左右，汽车中的计算机就会从发动机中的各个传感器读取一些关键的性能指示数据，并对汽车进行微调以确保运转正常。该功能的关键至少有某些部分是在传感器中，计算机本身只处理电信号。传感器负责确定发动机究竟运转状况如何，并将这些信息转换成一组电信号，用以描述或表示发动机的当前状态。类似，计算机所做的调节被存储为电信号，并被转换成为发动机工作状况的实际变化。

电信号如何能“表示”信息？计算机如何精确地处理这些信号，以达到精细的控制而无需任何人的干涉？这种表示问题就是理解计算机如何工作以及如何在现实世界中部署计算机的关键。

1.1.2　算法机

计算机操作方面的最重要的概念就是算法（algorithm）：算法就是一个明确的、按步进行

的过程，用以解决某个问题或达到某个期望目标。计算机的最终定义不依赖于其物理特性，甚至也不依赖于其电学特性(如其晶体管)，而是依赖于其表示和完成算法的能力，这些算法来源于存储的程序。在计算机内部是数以百万计的微小电路，每个电路在被调用时都执行一个特定的、明确定义的任务（如将两个整数相加、使单个或一组线通电）。大多数使用计算机或者为计算机编程的人都不知道这些电路的工作细节。

特别是，一个典型的计算机能执行若干个基本类型的操作。由于计算机从根本上看只是计算机器，所以它能执行的几乎所有功能都与数字（以及用数字表示的概念）相关。一个计算机通常能执行诸如加法和除法等基本的数学操作。它也能执行基本的比较操作，如一个数字与另一个数字相等吗？第一个数字小于第二个数字吗？它能存储几百万或几十亿的信息片断，并能单独地获取。最后，它能根据获取到的信息和执行比较的结果来调整其动作。比如，如果获取到的值大于以前的值，则说明发动机正在过热的情况下运行，需要发一个信号来调节其性能。

1.1.3 功能部件

系统级描述

几乎任何大学的公告板上都有这样一些广告，如“超级计算机！ 3.0-GHz Intel Celeron D, 512mg, 80-GB硬驱，15英寸显示器，为抵汽车款忍痛割爱！”。像大多数广告一样，许多信息需要深入地解读才能理解其全部意思。例如，15英寸显示器的哪个部分才真正是15英寸？（显示屏对角线的长度，够奇怪的了）。为了理解计算机的工作细节，我们首先必须理解主要的部件以及它们相互之间的关系（见图1-1）。

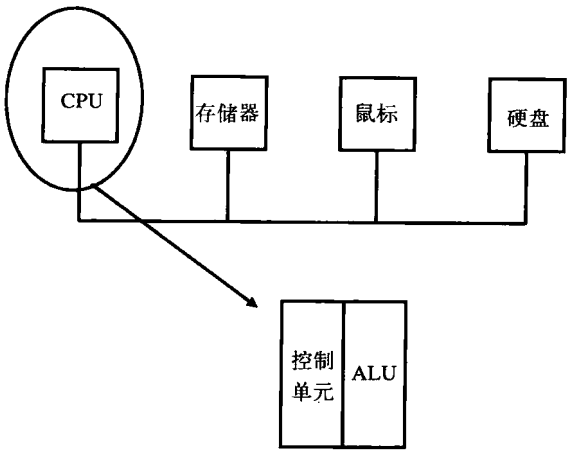


图1-1　一台计算机的主要硬件部件

中央处理单元

计算机的心脏就是中央处理单元（Central Processing Unit），即CPU。这通常是制造在单个集成电路（Integrated Circuit, IC）硅片上的一个高密度电路（见图1-2）。它通常看起来就像一小块硅片，安置在一个几平方厘米并由金属管脚包围着的塑料厚片上。塑料厚片本身座落在一个母板(motherboard)上，母板就是一个电子电路板，由一块塑料和金属组成，每个面几十平方厘米，包含了CPU和其他一些部件，这些部件因速度和便利等原因而需要安置在靠近CPU的地方。CPU是计算机的最终控制器，也是执行所有计算的地方。当然，这也是人们

谈论计算机时所指的部分，比如："3.60GHz奔腾4计算机，如惠普HP xw4200"，就是指这样的一台计算机：它的CPU是奔腾4芯片，运行速度为3.6千兆赫兹(GHz)，即每秒3 600 000 000个机器周期(machine cycle)。计算机的多数基本操作需要一个机器周期，所以另外一种描述方式就是3.60GHz计算机能在每秒执行超过35亿个基本操作。在写本书的时候，3.6GHz是速度很快的机器，但随着工艺进步，情况变化很快。例如，在2000年，1.0GHz奔腾是最先进的，根据计算能力每18个月翻一番这一长期证明有效的经验（摩尔定律），我们可以预测8GHz CPU在2008年将会普及。

图1-2 CPU芯片的照片

CPU通常以工艺发展的系列来描述：例如，奔腾4是奔腾、奔腾2及奔腾3的进一步发展，都是Intel公司生产的。在这之前，奔腾本身衍生于一长串用数字编号的Intel芯片，开始于Intel 8088，并发展到80286、80386及80486。这个所谓的"x86系列"就成为最畅销的IBM个人计算机（PC机及其模仿机）的基础，并且可能是最广泛使用的CPU芯片。现代苹果计算机采用了一个不同系列的芯片，即PowerPCG3和G4，该系列芯片由苹果、IBM及Motorola组成的联盟（AIM）所制造。较早的苹果和Sun工作站采用了Motorola设计的68000系列芯片。

CPU本身可分为两个或三个主要的功能部件。控制单元(Control Unit)负责在计算机内移动数据。例如，控制单元要从存储器中装入单个程序指令，辨别各指令的功能，并将指令传递到计算机的其他部分来执行。算术和逻辑单元（ALU）为计算机执行所有必需的算术运算。它通常包含完成加法、乘法、除法等的特殊用途硬件。顾名思义，它还执行所有的逻辑操作，确定一个给定的数是否大于或小于另外一个数，或者检查两个数是否相等。一些计算机（特别是较早的）有专用的硬件，有时安置在与CPU不同的芯片上，用于处理涉及分数和小数的操作。这个特殊的硬件通常称为浮点单元或FPU（也称为浮点处理器或FPP）。其他计算机将FPU硬件放在ALU和控制单元所在的同一个芯片上，但FPU仍可看作是同一电路系统内部的不同模块。

存储器

要执行的程序和其数据都存放在存储器中。在概念上，存储器可看作是一个非常长的一列或一排的电磁存储器件。这些列的位置从0到CPU定义的最大数进行编号，可由控制单元单独地寻址，将数据置入存储器或将数据从存储器中取回（图1-3）。此外，多数的现代计算机都允许像磁盘驱动器这样的高速设备拷贝大块的数据而无需对每个信号都要控制单元介入。

存储器可宽泛地分为两种类型：只读存储器（ROM），这是永久的、不可改变的，甚至在电源关闭后数据仍保留；随机存取存储器（RAM），其内容可被CPU改变，是作为临时存储器但通常在电源关闭后数据消失。很多机器中同时有这两种存储器。ROM保存标准化的数据和操作系统基本版本，用于启动机器。更多程序保存在像磁盘驱动器和CD这样的长期存储器中，并在需要时被装入RAM中用于短期存储和执行。

补充资料

摩尔定律

Intel的共同创始人戈登·摩尔在1965年观察到，能放置到一个芯片上的晶体管数量每年翻一番。在20世纪70年代，这个步伐稍微变慢了，成为每18个月翻一番。但令每个人包括摩尔博士本人吃惊的是，从此这个步伐就异乎寻常地均匀。仅仅在刚过去的几年这个步伐才减慢。

更小的晶体管（相应的晶体管密度更大）所蕴涵的意义是深远的。首先，硅片本身每平方英寸的成本比较而言已经相对稳定了，所以密度加倍就使芯片的成本近似减半。其次，更小的晶体管反应更快，并且部件能更紧密地放置在一起，所以它们能更快地相互通信，极大地提高了芯片速度。较小的晶体管也消耗较少的功耗，这意味着更长的电池寿命和较低的散热要求，避免了对房间气温控制及庞大电扇的需求。由于更多的晶体管可放置到一个芯片上，就需要较少的焊接以将芯片连接到一起，因而就降低了焊接破损的机会，相应地也就极大地提高了总体可靠性。最后，芯片更小的事实意味着计算机本身可以做得更小，小到足以使嵌入式控制芯片和/或个人数字助理（PDA）这些思想成为现实。很难预测摩尔定律对现代计算机的发展产生多大影响。到目前，摩尔定律通常就是简单地用来表明计算机的能力在每18个月翻一番（不管什么原因，不仅仅是晶体管密度）。一个当地商店下架的标准的、甚至低端的计算机就比1973年的原始Cary-1超级计算机更快、更可靠，并且有更多的内存。

摩尔定律的问题是它不会永远保持。最终，物理定律指出，晶体管不能小于一个原子（或类似的东西）。更令人不安的是摩尔第二定律指出制造成本在每3年翻一番。只要制造成本增加的速度比计算机能力增长的速度慢，性能/成本比就应该保持合理。但是，投资新的芯片工艺的成本很大，可能会使Intel这样的制造商继续投入新的资本变得困难。

这个简化的描述故意隐藏了存储器的一些繁杂方面，通常硬件和操作系统为用户处理这些方面。（这些问题也趋向于与硬件相关，所以将在后面章节详细讨论。）例如，不同的计算机，即使有相同的CPU，也常常有不同数量的存储器。安置在计算机上的物理存储器可少于CPU能寻址的最多位置数，但在特定情况下，却可能多于CPU能寻址的最多位置数。进一步地，处于CPU芯片本身的存储器通常可以比处于不同芯片上的存储器以快得多的速度访问，所以，一个好的系统应努力确保数据需要移动或拷贝时能在最快的存储器中得到。

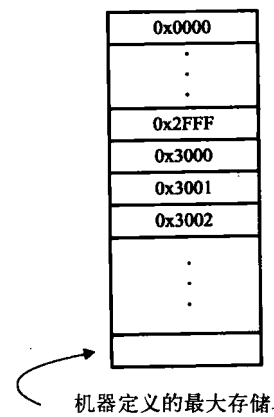


图1-3　线性排列的存储器单元的示意图

输入/输出（I/O）外设

除了CPU和存储器，一台计算机通常还包含其他

设备用来读、显示或存储数据，或者更一般地与外部世界交互作用。这些设备多种多样，从普通的键盘和硬盘驱动器，到并不普通的设备如传真（FAX）板、话筒及音乐键盘，再到相当怪异的小配件像化学传感器、机器人手臂及安全门拴。这些部件的一般术语就是外设(peripheral)。在很大程度上，这些设备对计算机本身的体系结构和组织几乎没有直接的影响，它们只是信息的源点和终点。例如，键盘只是一个让计算机从用户那里获取信息的设备。从CPU设计者的角度看，数据就是数据，不管它来自于因特网，来自于键盘，还是来自于奇特的化学频谱分析仪。

在很多情况下，一个外设可在物理上分为两个或更多的部分。例如，计算机通常在某种形式的视频监视器上将其信息显示给用户。监视器（monitor）本身就是一个独立的设备，通过电缆连接到计算机机箱内部的视频适配板上。CPU画图时，可发送命令信号给视频板，视频板生成图形并通过视频电缆将适当的视觉信号发送到监视器本身。可以用类似的过程描述计算机如何通过一个SCSI（Small Computer System Interface）控制器卡从很多不同种类的硬盘驱动器装入一个文件，或者通过一个以太网卡与构成因特网的数百万英里的线交互作用。在概念上，工程师们会对设备本身、设备电缆（通常只是一条线）及设备控制器（通常是计算机内部的一块板）加以区分，但对于程序员，它们通常从总体上就被看作是一个设备。根据这种逻辑，整个因特网连同其数百万英里的线就只是“一个设备”。对于一个设计良好的系统，从因特网下载一个文件与从一个硬盘驱动器装入一个文件之间没有多大区别。

互连和总线

为了使数据在CPU、存储器以及外设之间移动，必须存在连接。这些连接（特别是独立的板之间的连接）通常是成组的线，使得多个单独的信号能成组发送。例如，最初的IBM-PC有8条线在CPU和外设之间传递数据。一台更现代的计算机的PCI（Peripheral Component Interconnect）总线有64条数据线，即使不考虑计算机速度的提高，也能使数据以8倍的速率传递。这些线通常组合起来形成总线（bus），总线就是连接若干不同设备的线的集合。由于总线是共享的（就像早期的共线电话），一次只有一个设备能传送数据，但连接的所有设备都可得到数据。一些附加信号用于确定哪个设备应该得到数据以及当它得到数据时应该做些什么。

一般来说，连接到单个总线的设备越多，运行就越慢。这有两个原因。首先，设备越多，在相同时间两个设备要同时传送数据的可能性就越大，因此一个设备就要等待轮到自己传送。其次，更多的设备通常意味着总线更长，由于传播有延迟（即信号从线的一端到达另一端的时间），因而就降低了总线的速度。由于这个原因，很多计算机现在采取多总线设计，例如，局部总线连接CPU和CPU母板上的高速存储器（通常置于CPU芯片本身），系统总线连接存储器板、CPU母板以及一个“扩展总线”接口板，而扩展总线又是一个连接网络、磁盘驱动器、键盘及鼠标的二级总线。

在个别的高性能计算机上（如图1-4所示的计算机），可能有4到5条独立的总线，一条保留用于高速、数据密集的设备，如网络和视频卡，而低速设备如键盘则归入另外的低速总线处理。

支持单元

除了已经提到的设备，一个典型的计算机还有一些对保证其物理条件很重要的部件。例如，在机箱内部（机箱本身也是对易损电路板的重要物理保护）有一个供电电源，用于将交流电压转换成可用于电路板的适当调节的直流电压。还可能有一个电池，尤其在便携式电脑中，用于在没有墙壁插座提供电流时提供电源并保持存储器的设置。通常还有一个风扇用来

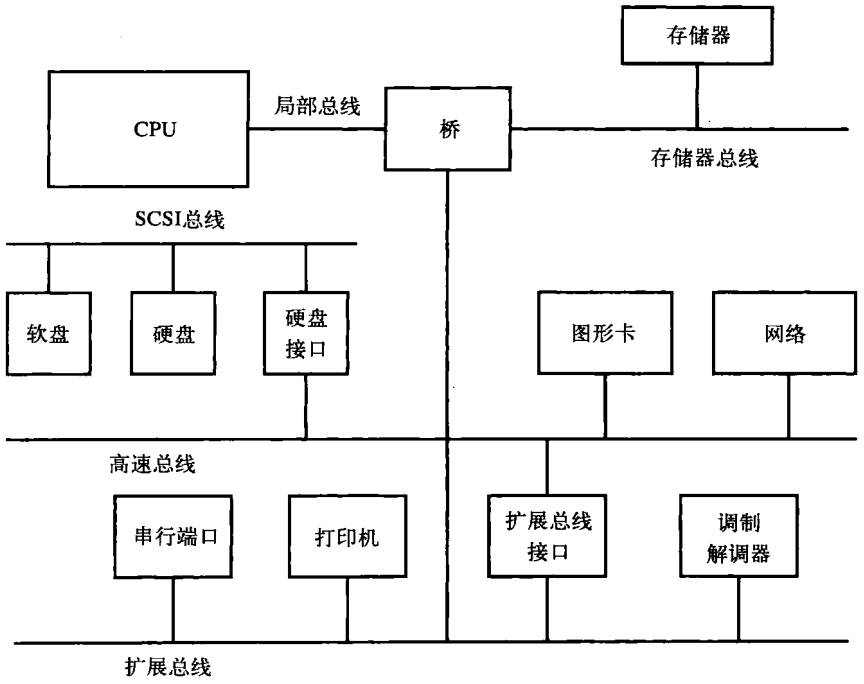


图1-4　高速多总线计算机的体系结构示意图

对机箱内部通风以防止部件过热。可能还有其他一些设备，如热传感器（用于控制风扇速度）、用于防止未授权的使用和拆卸的安全设备，常常还有若干个完全内部使用的外设如内部磁盘驱动器和CD阅读器。

1.2　数字和数值表示

1.2.1　数字表示和位

在最基本的层次，计算机部件像很多其他电子部件一样有两个状态。灯或者开或者关，开关或者打开或者关闭，线路或者正携带电流或者未携带电流。对于计算机硬件，各个部件如晶体管和电阻器相对于地或者处于0电压或者处于某个其他电压（典型值是对地有5伏特电压）。这两个状态通常用来分别表示数字1和0。在计算机发展的早期阶段，这些值是通过翻转机械开关来进行手工编码的。现在，高速晶体管实现的是几乎同样的意图，但数据用这两个值来表示自上世纪四十年代以来一直未改变。每个这样的1或0通常称为位或比特（bit），是“binary digit”的简写。（当然，“bit”本身是一个标准的英文词，意思是“一个很小的量”，这也描述了“一小部分”的信息）。

补充资料

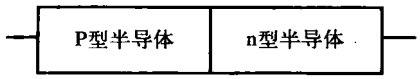
晶体管如何工作

在现代计算机中最重要的电子部件是晶体管，晶体管是由Bardeen、Brattain和Shockley于1947在贝尔电话实验室发明的。（这些人因为这项发明而获得了1956年的诺贝尔物理学奖）。它的基本思想涉及一些相当高级的（是的，诺贝尔档次的）量子物理学，但是，不需要具体的公式，你也能根据电子迁移的原理来理解它。一个晶体管主要包含了

一种称为半导体（semiconductor）的材料，它处于好导体（如铜）和坏导体/好绝缘体（如玻璃）之间的不稳定的中间状态。半导体的一个关键方面是其导电能力会随半导体中杂质（掺杂物，dopant）的不同而显著变化。

例如，当元素磷被加入到纯硅（一种半导体）时，将为硅提供电子。由于电子带负电，磷就称为n型（n-type）掺杂物，而用磷掺杂过的硅就叫做n型半导体（n-type semiconductor）。与此相反，铝是一种p型（p-type）掺杂物，从硅阵列中移出（实际上是锁定）电子。在p型半导体中，这些电子被移出的地方有时称为“空穴”。

当将一块n型半导体与一块p型半导体紧挨着放在一起时（所形成的小器件称为二极管(diode)，见图1-5），就会发生有趣的电效应。电流一般不能穿过这样的二极管，载流的电子将遭遇并“陷入”空穴。然而，如果你将一个偏置电压（bias voltage）施加于这个器件，额外的电子将填充空穴，使得电流通过。这意味着电流只能在一个方向穿过二极管，使得其具有电整流器（rectifier）的用途。

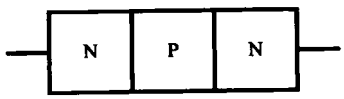


二级管的结构

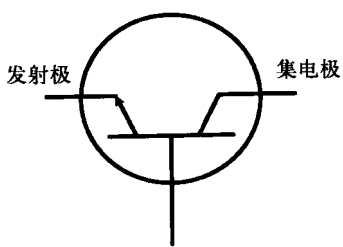
二级管的符号

图1-5 二级管

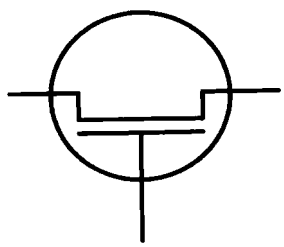
现代的晶体管做得就像一个半导体三明治，就是一个p型半导体薄层处于两薄片的n型半导体之间。(是的，这就是两个背靠背的二极管，见图1-6)。在常态环境下，电流不能从发射极(emitter)到集电极（collector）穿过，因为电子陷入到空穴中。将偏置电压施加到基极（base）(中间的线）就将填充空穴使得电流能通过。你可以将基极看作一个可开关的门，使得电流能流过或不能通过。或者，你也可以将其看作是橡胶软管中的一个阀门，用以控制通过的水流量。将阀门转到一个方向，电信号就减小到很微弱；转到另一个方向，就无阻碍地流动。



NPN晶体管



晶体管符号



晶体管的另一种符号

图1-6 晶体管