石教英等 编

Computer

计算机体系结构

浙江大学出版社

图书在版编目（CIP）数据

计算机体系结构 / 石教英等编. —杭州：浙江大学出版社，1998.10(2002 重印)
ISBN 7-308-02055-X

Ⅰ. 计... Ⅱ. 石... Ⅲ. 计算机体系结构
Ⅳ. TP303

中国版本图书馆 CIP 数据核字（2001）第 095113 号

责任编辑 李玲如
出版发行 浙江大学出版社
（杭州浙大路 38 号 邮政编码 310027）
（网址：http://www.zjupress.com）
（E-mail：zupress@mail.hz.zj.cn）
排　　版 浙江大学出版社电脑排版中心
印　　刷 浙江省良渚印刷厂
经　　销 浙江省新华书店
开　　本 787mm×1092mm 1/16
印　　张 28.5
字　　数 730 千字
版　　次 1998 年 10 月第 1 版
印　　次 2006 年 8 月第 7 次
印　　数 12001—13500
书　　号 ISBN 7-308-02055-X/TP・175
定　　价 30.00 元

前　言

近十年来，计算机发生了巨大变化，这一变化反映在计算机的硬件、软件和应用各个方面，并且变化速度之快不仅使计算机用户，而且也使计算机软硬件设计人员感到应接不暇，稍有不慎便会感到落伍。以使用高性能微处理器为代表的工作站的性能以每 3 年增加 4 倍的速度发展。1992 年工作站的性能约为 50 SPEC(SPEC 为性能测试基准)，1995 年已达到 200～300 SPEC，2000 年预计将达到 1 000 SPEC。工作站和个人机的速度越来越快，体积越做越小，而价格却越来越便宜，极大地推动了计算机应用事业的普及。促进计算机出现上述变化的原因是多方面的，主要可归纳为计算机实现技术和系统结构技术的进步。计算机实现技术的进步主要体现在：逻辑集成电路的晶体管密度以每年 60%～80%的速度递增，运算速度每年也以同样速度递增；DRAM(动态随机存取存储器)的密度以每年 60%的速度递增；磁盘的容量每年以 50%的速度递增(而 1990 年以前，每年递增速度仅为 25%)。计算机系统结构技术在 80 年代 RISC(精简指令集计算机)技术成功的基础上，又在硬件支持和编译支持的指令级并行性开发技术，多发射微处理器(超级标量和超长指令字)技术；采用 2 级、3 级 Cache 匹配处理器速度技术；冗余磁盘陈列(RAID)技术；存储共享多处理器技术以及工作站网络并行计算技术等方面取得重大进展。正是以上技术进步保证了单处理器计算机性能的继续高速发展，并保持其在计算机市场上的主流地位。另一方面，在可以预见的 5～10 年内，小规模(由 4，8 或 16 个微处理器组成)多处理器系统将集成到个人计算机平台。

计算机系统结构是计算机科学与技术领域的重要学科，也是高等院校计算机系本科生和研究生学习设计、分析和评价计算机的主干课程。与上述计算机本身高速发展形成鲜明对照的是目前计算机系统结构课程教学的内容和方法都显得落后，急需进行改革。首先要改革的是教材，要有一本能较好反映近年来计算机发展的好教材。近年来，在国内找不到适用教材的情况下，我们浙江大学、杭州大学和杭州电子工学院三校计算机系于 1994 年开始采用美国斯坦福大学 John L. Hennessy 和加州大学贝克利分校 David A. Patterson 编著的《计算机系统结构——定量方法》的原版教材进行计算机系统结构课程的教学。由于教材内容新颖，同时配备了相应上机实践，改变了空洞、枯燥的教学方法，受到了同学们的欢迎。在三年教学实践的基础上，我们三校联合编写这本《计算机体系结构》教程。我们的宗旨是：

1. 教材要尽可能反映计算机系统结构领域的最新成果，让同学们明白近十年来计算机性能突飞猛进的原因，激发他(她)们的学习热情和钻研精神。

2. 以讲授计算机系统结构的基本概念和基本原理为主，而不是完整地介绍各种系统结构，即不是以具体机器为实例进行教学。

3. 引进定量原理，让学生学会如何测试实际机器，分析实际机器，分析计算机设计中遇到的各种限制因素，培养正确选择各种折衷方案的能力。

4. 强调计算机系统结构与操作系统和编译系统的相互关系，充分反映出计算机系统结构不是单纯的硬件课程，而是硬件和系统软件的结合点，因此本课程不仅适用于培养系统结构和芯片设计工程师和计算机系统工程师，而且也适用于培养编译系统和操作系统工程师。

全书共分八章，第一章“计算机设计基础”，介绍计算机设计者的任务、计算机性能评价和

计算机成本的构成。第二章“指令集的设计”，重点在于介绍指令集的设计原则、操作系统和编译系统与系统结构的相互关系、指令系统的测量方法，最后简要介绍 RISC 的设计思想，第三章“CPU 的设计”，以 RISC 样机指令集为例介绍处理器数据通路和控制器设计，以及中断系统。第四章“流水线技术”内容包括流水线技术基础和高级流水线技术两大部分。基础技术包括流水线处理的基本原理和流水线的动态调度、指令级并行性的开发和非线性流水线等内容。第五章“存储器层次结构”，包括存储器层次结构的基本概念、Cache、主存和虚存等基础部分和改进 Cache/主存性能的新技术，最后以 Alpha 机存储系统为实例综合介绍存储系统的工程过程。第六章“计算机输入/输出系统”介绍输入/输出设备类型、输入/输出子系统的控制方法、总线、输入/输出子系统性能测量和冗余磁盘阵列(RAID)技术。第七章“网络并行计算系统”介绍应用通用互连网络和商品化的工作站构成并行计算系统的硬软件技术，包括互连网络原理、工作站网络并行计算系统实例(NOW 计划)介绍和实现网络并行计算的系统软件技术。第八章“多处理器计算机结构”介绍当前研究热点：共享存储器系统结构的多处理器系统，包括集中共享存储器和分布共享存储器系统结构、同步机制和存储器一致性模型等内容。

从以上介绍可以看出，本书的内容以介绍单处理器计算机系统结构为主，多处理器计算机系统结构为辅，这样安排是为了充分反映目前单处理器计算机在技术上持续发展的成功，在市场上不可动摇的主流地位，以及能满足 99%应用需要的事实。在具体内容取舍上，我们把重点放在基础部分，着重介绍基本概念，同时也顾及那些促进微处理性能突飞猛进的新技术。总的讲，本书材料超出本科生“计算机系统结构”一学期课程的内容，任课教师可根据实际情况运用本书的材料，在讲清楚基本内容的前提下，适当介绍相关新技术；可以将有关最新进展的内容作为学生课外阅读材料，也可留作硕士研究生相关课程的材料。

本书编写分工是：第一、二、四章由施青松执笔，第三、六章由蒋纯执笔，第五章由姜晓红执笔，第七章由石教英和蒋纯执笔，第八章由石教英和胡维华执笔，全书由石教英修改定稿。

由于作者水平有限，书中难免有不妥之处，恳请读者不吝赐教。

石教英

1997 年 12 月 30 日

目　录

第一章　计算机设计基础

一个计算机设计者首先应该知道：设计一台计算机应经过哪几个过程，首先应做什么工作，一台计算机应有怎样的功能，如何评价这些功能，以及一台计算机的成本构成。这一章就是向读者介绍这方面的知识。

1.1　计算机设计者的任务

计算机设计者的工作是多方面的，包含有指令集设计、功能组成设计、逻辑电路设计和硬件结构的设计等。但是我们认为在作上述工作之前，首先应该知道计算机应具有的性能是什么、这些性能的选取原则是什么以及如何优化实现。

1.1.1　计算机需求的功能

1.1.1.1　什么是计算机需求的功能

计算机需求的功能实际上就是用户需求的功能。这些功能需求来自计算机应用的各个领域，它们之中有些功能是一致的，而有的功能是根本不同的。表 1-1 列举、归纳了计算机最主要的功能需求和它们所应支持的特征。这些功能需求中的大部分是最基本的功能。例如，现代操作系统中使用虚拟存储器和保护功能是很普遍的，这就必须为它建立最基本的硬件支持，否则机器就不能很好地工作。但是有些功能的实现是高成本的或现代技术难以达到的。那么，对于计算机的这些需求，我们又如何来处理和实现呢？这就是下面要讨论的问题。

1.1.1.2　计算机应具有的性能

我们已经了解了计算机的用户需求。满足这些需求是体系结构设计者努力追求的目标。从前一节我们已经知道，并不是所有需求的功能都能轻而易举地实现，因此对这些需求，我们要进行筛选。筛选的依据是**成本性能比**，这是计算机体系结构设计的最基本原则。对于那些基本的功能要求，尤其是与已有计算机兼容的问题，一般总是给予满足。但对于实现成本高，或现代技术还难以实现的功能，我们就不是直接给予满足，而是将其分解，间接实现。也就是说部分直接实现，部分给予间接实现。这里所说的直接实现一般指用硬件实现，而间接实现是指用软件实现。区分的界线就是成本性能比和用户的承受能力。例如，一种极端情况是对于某些特殊行业或科研性的设计，成本可以不计，但性能必须给予保证。

表 1-1　计算机的主要功能需求和应支持的典型特征

功能需求	应支持的典型特征
应用领域	采用计算机的结构特征
特殊目的	特殊应用的高性能计算机
通用目的	任务和性能平衡的计算机
科学计算	高性能浮点计算机
商业应用	支持 COBLE(十进制运算)、支持数据库和事务处理
软件层的兼容	由计算机现有软件量决定
高层语言	设计者有最大的灵活性,但需要新的编译
目标码或二进制兼容	完全由结构决定,没有灵活性
操作系统(OS)需求	支持操作系统必要的特性
寻址空间范围	非常重要的特征:可能限制应用
存储器管理保护	现代操作系统必需:用于分页、分段管理等
	不同 OS 和应用的需要:页虚拟存储器,段保护
标准	规范计算机市场需用的标准
浮点	浮点格式和运算:IEEE,DEC,IBM
I/O 总线	用于 I/O 设备:VME,SCSI,NuBus,Futurebus
操作系统	UNIX,DOS 或销售商专用操作系统
网络	需要支持不同网络:Ethernet,FDDI,ATM
高层语言	语言影响指令集:ANSI C,FORTRAN 77,ANSI COBOL

因此,我们设计计算机的功能并不就是用户所需要的功能,它要考虑到成本、技术、兼容性和市场大小等诸多因素,体现在设计和生产中要有赢利,用户要承受得了。衡量原则就是成本性能比。根据这一原则,我们首先要设计确定计算机应具有的功能,我们也称之为**功能结构设计**。完成了这一步,我们就可以进行指令结构和工艺结构的设计。

为了充分优化功能结构设计,下面几节着重介绍计算机设计的几个定律和原则,以及计算机性能评价和计算机成本构成等。

1.1.2　功能实现时软件和硬件的选择

我们已经知道,某些需求功能用硬件来实现并不十分合算,因而可以用软件来间接实现。那么它们各自有什么优缺点,如何选择,衡量的尺度又是什么?

1.1.2.1　软件实现的优点

软件实现的优点是显而易见的,那就是低成本、易设计、低出错率、可改性强、适用性强、设计周期短以及升级提高较简单。但是最大的缺点是执行速度慢,一般是以牺牲时间来实现其功能的;另外,相对的存储器费用和软件设计的费用也要增加。但是软件实现并不一定是最慢的,有时,一个好的软件算法要比一个较差的硬件实现的算法更快,特别是近几年强调编译的重要性,使得软件的算法变得越来越重要。例如 RISC(精简指令集计算机)技术的计算机就特别强调针对机器特性进行编译。

1.1.2.2 硬件实现的优点

对于某一功能来说，我们一般都希望直接用硬件来实现，这样可以提高计算机在这一方面的性能(速度)。如果全部能用硬件来实现计算机的功能，那么计算机的性能也许可以提高很多，但是它的造价是十分昂贵的。由于技术和工艺等条件的限制，使得这样的设计周期很长，致使市场适应能力很差。例如某一功能的实现花费了三年的时间，那么三年前设计的结构，在三年后可能已经落后了，这显然是无效劳动。但是有些功能必须用硬件支持才能很好地工作。例如用于科学计算的计算机，其浮点运算是必不可少的，而且也很常用，对于这种情况使用硬件实现浮点运算是必要的。又例如商用计算机要频繁使用十进制和字符串的运算功能，那么用硬件实现这些功能就十分有效。那么，对一台计算机需求的功能来说，到底哪些用软件实现，哪些用硬件实现呢？一句话：软件和硬件要综合考虑，要以价格性能比高低为取舍原则。

1.1.2.3 软件和硬件实现的取舍原则

我们首先来看一下近四十年软件和硬件的价格发展趋势(见图 1-1)。可以看出早期曾是免费的软件如今随着其功能的日趋复杂价格呈稳定的上升趋势；另一方面，在同一时期内随着半导体和计算机技术的发展，硬件成本以难以置信的速度下降。这一点我们可以从今天的计算机市场上看到，1997 年还要一万左右人民币一台的 Pentium 多媒体微型计算机，今天只要五六千人民币就能买到，明天或许还要低。

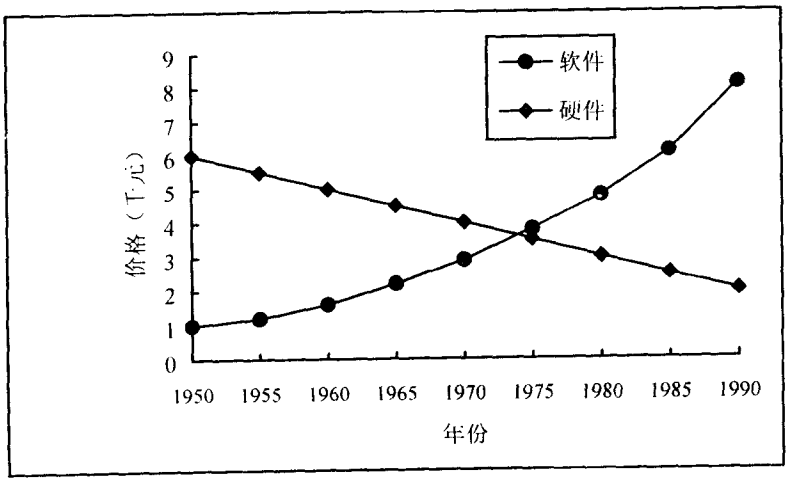

图 1-1 软件和硬件价格变化趋势

从图 1-1 中的曲线上我们可以看到一个市场价格的发展趋势。硬件成本在下降，那么是不是硬件成分越多越好呢。实际上，软件和硬件的价格组成虽然复杂，但都由两部分组成：

1. 研制成本。
2. 重复制造成本。

我们假设某一功能用硬件实现的研制成本为 D_h，软件实现的研制成本为 D_s。在目前的条

件下硬件的设计费用还是明显地大于软件的设计费用。设$D_h = nD_s$，则一般$n > 100$。在重复制造过程中，硬件的重复生产实现也要比软件的重复生产贵得多。软件重复生产只要拷贝费用加上存放程序的存储介质（如软盘）的价格。硬件生产则要包含所有器件成本费用。这和初始生产时的费用相近，只是除掉了设计开发等费用。因此，我们假设某功能用软件实现时重复生产费用是M_s，而用硬件实现时重复生产的费用为M_h。那么M_h远远大于M_s，如$M_h > 100M_s$是完全正常的。设

$$M_h = mM_s$$

进一步分析，发现用硬件实现某一功能（如子程序调用的全部操作）往往只需设计一次，而用软件实现时，每用到该功能往往需要重新设计一次，只不过设计费用要比第一次设计时的D_s低得多，只是第一次设计的简单修改和复制。假设该功能用软件实现时需重新设计C次，则该功能用软件实现的总研制费用为$C \times D_s$。同一功能的软件在存储介质上有可能出现多次，每出现一次都要增加拷贝和存储的费用。设该功能在存储介质上出现了R次。则软件实现该功能时重复生产的费用为$R \times M_s$。

我们进一步假设，根据市场需求，具有该功能的计算机共生产了V台，则硬件实现该功能时每台的费用为

$$\frac{D_h}{V} + M_h$$

用软件实现该功能每台的费用为

$$\frac{CD_s}{V} + RM_s$$

一般只有当硬件成本小于等于软件成本时用硬件实现才较为合适，即

$$\frac{D_h}{V} + M_h \leqslant \frac{CD_s}{V} + RM_s$$

用$D_h = nD_s$和$M_h = mM_s$代入上式后可以得出

$$\frac{nD_s}{V} + mM_s \leqslant \frac{CD_s}{V} + RM_s$$

从公式中我们可以看出在两种情况下不等式可以成立：

第一种情况是C和R的值较大，即这个功能用得很频繁，用到的次数很多，是一个常用的基本功能。这种情况用硬件实现较合适。这个结论和后面我们要介绍的Amdahl定律的结果是一致的。这说明并不是硬件比例越多越好。

第二种情况是V值很大。在这里我们假设$D_s = pM_s$。因为软件的研制成本远大于其重复生产成本，则

$$\frac{npM_s}{V} + mM_s \leqslant \frac{CpM_s}{V} + RM_s$$

$$\frac{np}{V} + m \leqslant \frac{Cp}{V} + R$$

一般C值总是远小于n，那么只要V值越大这个不等式成立的可能性越大。也就是说，只有产量大的计算机系统，某些功能用硬件实现才适宜。

从上述分析我们得出的结论是，常用的基本功能或产量很大的功能才适宜于用硬件实现。但是这里仅仅考虑了成本因素。另外还有一个很重要的性能因素被忽略了。我们应该考虑要用硬件实现的功能较软件实现有多大的性能提高，也就是投入的硬件实现经费和提高的性能是

否能被市场接受。用户有一种少增加投入多获得效益的消费心理。能做到这一点的话，市场会增大，V 值也会增大；产量大成本就会下降。设计者最终要用成本性能比作为软、硬件实现功能的取舍标准；消费者要用价格性能比作为选购计算机系统的取舍标准。

最后归纳几点作为设计人员的参考原则。这些建议也不是不变的，随着技术工艺的发展会不断充实变更。

1. 考虑用户的应用领域：由于硬件的适应性和灵活性较差，对于功能单一的专用计算机，如控制计算机、专用于传感器模式识别的计算机和算法单一的智能仪器等可以适当考虑用硬件实现。

2. 设计周期长的硬件不宜采用：由于计算机技术发展很快，太长的设计周期会导致新机器一出世就被淘汰。而硬件实现的设计周期本来就长，且工艺技术要求又很高，如某一功能由于硬件设计周期长而没有了市场竞争力，那就不适宜用硬件实现。

3. 常用的功能应尽量采用硬件实现。

4. 实现功能的成本性能比（或价格性能比）要低：既要为生产商接受又要为用户接受。

5. 超前设计：计算机设计总有一个设计周期，要保证投放市场后具有先进性和竞争力，必须在设计新的计算机结构时注意计算机及相关技术的发展动态，尽量采用最新的技术和最新的工艺以及超前的设计思想和适当超前的技术和工艺。

这一节的软、硬件实现功能的取舍原则对系统结构的设计者来说是很重要的，完成了这一功能结构设计后，下一步就要进入指令结构的设计。

为了加深理解，我们举几个例子。

［例 1-1］ 某一计算机用于商业外贸的事务处理，有大量的字符串处理操作。由于这种商务处理很普遍，有较大的市场，故而设计人员决定在下一代此类计算机的 CPU 中加入字符串操作的功能。经测试应用软件调查发现，字符串操作的使用占整个程序运行时间的 50%。而增加此功能如用软件（如微程序）实现，则快 5 倍，增加 CPU 成本 1/5 倍；如果用硬件实现，则快 100 倍，CPU 成本增加到 5 倍。问设计人员提出增加此功能是否恰当？如恰当则此功能应该用软件实现还是用硬件实现？设 CPU 成本占整机成本的 1/3。

解：首先来计算机器在两种情况下提高的性能和成本性能比。

设：S 为 CPU 未增加字符串功能时的 CPU 平均速度，T_{old} 为此时运行程序的时间，T_{new} 为增加字符串功能后程序运行的时间，则

1. 硬件实现

$$T_{new}=T_{old}(1-50\%)+\frac{50\%T_{old}}{100}$$

$$\text{性能变化}=\frac{T_{old}}{T_{new}}=\frac{T_{old}}{T_{old}(1-50\%)+\frac{50\%T_{old}}{100}}=\frac{1}{1-50\%+\frac{50\%}{100}}=1.98\text{ 倍}$$

$$\text{成本增加}=\frac{2}{3}\times 1+\frac{1}{3}\times 5=2.33\text{ 倍}$$

$$\text{成本性能比}=\frac{2.33}{1.98}=1.18\text{ 倍}$$

2. 软件实现

$$T_{new}=T_{old}(1-50\%)+\frac{50\%T_{old}}{5}$$

$$性能变化=\frac{T_{old}}{T_{new}}=\frac{T_{old}}{T_{old}(1-50\%)+\frac{50\%T_{old}}{5}}=\frac{1}{1-50\%+\frac{50\%}{5}}=1.66\text{ 倍}$$

$$成本增加=\frac{2}{3}\times 1+\frac{1}{3}\times(1+\frac{1}{5})=1.07\text{ 倍}$$

$$成本性能比=\frac{1.07}{1.66}=0.64\text{ 倍}$$

从上面的计算分析可以得出增加字符串操作功能提高了整机的性能，两种方法均提高性能，且程度相近。但用硬件实现时成本性能比增加了 0.18 倍，而用软件实现时成本性能比却下降了 0.36 倍。因此设计人员提出增加字符串操作功能是恰当的，并且用软件实现此功能较好。

[例 1-2] 如果[例 1-2]中字符串操作功能的使用时间占整个程序运行时间的 90%，则情况又如何？

解：

1. 硬件实现

$$性能变化=\frac{T_{old}}{T_{new}}=\frac{T_{old}}{T_{old}(1-90\%)+\frac{90\%T_{old}}{100}}=\frac{1}{1-90\%+\frac{90\%}{100}}=9.17\text{ 倍}$$

$$成本性能比=\frac{2.33}{9.17}=0.25\text{ 倍}$$

2. 软件实现

$$性能变化=\frac{T_{old}}{T_{new}}=\frac{T_{old}}{T_{old}(1-90\%)+\frac{90\%T_{old}}{5}}=\frac{1}{1-90\%+\frac{90\%}{5}}=3.57\text{ 倍}$$

$$成本性能比=\frac{1.07}{3.57}=0.30\text{ 倍}$$

在这种情况下用硬件实现的成本性能比下降了 0.75 倍，软件实现的成本性能比下降了 0.70 倍。由于软件的成本通常远小于硬件成本，可能此时的硬件成本还是比较高，但是硬件实现的性能(9.17 倍)比软件实现的性能(3.57 倍)要高得多(9.17/3.57=2.57 倍)，因此用硬件实现更为合适。这样的结果和我们前面分析得出的 C 和 R 值较大时用硬件实现较合适的结论是一致的。

在上面两个例题中读者是否能归纳出部件和整机性能提高的内在规律。这里先请读者思考。下一节我们将要介绍这方面的定律。

1.1.3 计算机设计的几个原则

1.1.3.1 Amdahl 定律

在前面的例题中我们发现，尽管某一部分的功能性能被提高，但整机性能的提高远远小于部件性能的提高，并且整机性能的提高与被加速的这个功能在原机器的应用中所占的时间百分比有关。我们将前面例题的某些结果列于表 1-2。

从表 1-2 中我们可以得出，某部件应用越频繁，当提高该部件性能时，整机性能也提高得越多。但不管该部件性能提高(加速)多大(5 倍或 100 倍)，整机的性能加速不可能大于在原机器中除该部件外所有其他部件运行时间的百分比的倒数 $1/(1-F)$。

表 1-2　部件利用率和加速比关系

部件功能应用的时间百分比(F)	部件性能提高 5 倍时整机性能变化	部件性能提高 100 倍时整机性能变化
30%	1.32 倍$<\frac{1}{1-0.3}$	1.42 倍$<\frac{1}{1-0.3}$
50%	1.66 倍$<\frac{1}{1-0.5}$	1.98 倍$<\frac{1}{1-0.5}$
90%	3.57 倍$<\frac{1}{1-0.9}$	9.17 倍$<\frac{1}{1-0.9}$

上面归纳出的两点结论就是 Amdahl 定律。这个定律可以概括为：计算机性能的改善程度受其采用的快速部件(被提高性能的部件)在原任务中使用所占的时间百分比的限制。

为了说明计算机性能改善的程度与所采用部件性能提高程度之间的关系，我们可以将 Amdahl 定律归纳为定量的形式

$$\text{Speedup}=\frac{\text{提高部件性能后计算机的整机性能}}{\text{原计算机的整机性能}}=\frac{\text{原计算机完成一个任务的时间}}{\text{提高部件性能后计算机完成同个任务的时间}}=\frac{T_{\text{old}}}{T_{\text{new}}}$$

上述定义还不能看出采用先进的快速部件如何提高计算机的整机性能。下面我们通过日常生活中的例子来说明这个过程，并将加速比(Speedup)进一步具体化、数学公式化，使我们能清楚 Amdahl 定律的实质。

[例 1-3]　研究下面的春游问题。全体成员从杭州少年宫广场出发步行至初阳台，然后由此上山，翻山后再从岳坟下山，总共花费了 1 h。下山后再环湖一周回到出发地少年宫，共有 10 km 路程。如果第二段路程可以借用交通工具，则全程花费了多少时间？那么相对于全程步行，它们的加速比是多少？设步行的速度为 4 km/h，可以采用的交通工具有：

1. 普通自行车 A，速度为 10 km/h。
2. 赛车自行车 B，速度为 20 km/h。
3. 普通小轿车 C，速度为 50 km/h。
4. 赛车小轿车 D，速度为 300 km/h。
5. 新交通工具 E，速度为 400 km/h。

解：完成全程旅行分成两部分进行。首先要计算出第二段路程的时间，再加上第一段路程所花费的时间 1 h，才是完成全程旅行的时间。这是一个简单的算术问题，结果列于表 1-3。

从表中我们可以看出全程的加速比受到爬山时的 1 h 限制。虽然单项交通工具的加速比很高(100)但全程的加速比也只有 3.414 6，也就是说全程的加速比和被加速的路程(第二段路程)在全程中(步行时)的比例有关。如果用时间表示，我们称之加速的时间百分比，用 F 表示，如步行时 $F=2.5/3.5$，它始终小于 1。当然也和交通工具的加速比有关，我们用 S 来表示，如 $S=10/4=2.5$(普通自行车)。我们又设全程旅行加速比为 Speedup、步行完成全程旅行所需时间为 T_{old}、采用高速交通工具完成全程旅行所需时间为 T_{new}，则

表 1-3　各种运行工具的工作过程

交通工具	交通工具加速比	环湖一周时间 t $F=t/3.5$	全程旅行时间 T	全程旅行加速比 Speedup$=3.5/T$
步行	1.0	2.5 h	3.5 h	1.0
自行车 A	2.5	1.0 h	2.0 h	1.75
自行车 B	5.0	0.5 h	1.5 h	2.33
小轿车 C	12.5	0.2 h	1.2 h	2.9
小轿车 D	75	0.03 h	1.03 h	3.4
新交通 E	100	0.025 h	1.025 h	3.4146

$$T_{new}=T_{old}(1-F)+\frac{T_{old}F}{S}=T_{old}\left[(1-F)+\frac{F}{S}\right]$$

$$\text{Speedup}=\frac{T_{old}}{T_{new}}=\frac{1}{(1-F)+\frac{F}{S}}$$

根据上述推导的公式，可以完成表 1-3 的数据。

在这个例子中，为了缩短旅行时间，可以对交通工具进行改进，提高它的性能。如果将整个旅行作为一个计算机，而交通工具(步行、骑车、开车)等作为计算机的部件，那么上述问题和计算机整机性能与部件性能的关系是一样的。因此，我们可以将前述计算机整机性能加速比的定义公式变换成上述一般的数学形式。

这里 F 定义为采用先进高速部件的那部分程序在未采用先进高速部件的计算机上运行的时间占总时间的百分比，则

$$F=\frac{\text{采用高速部件的任务在老计算机上运行的时间}}{\text{整个任务在老计算机上运行的时间}}$$

同时将 S 定义为先进高速部件与老部件的性能比，则

$$S=\frac{\text{老部件完成该功能的时间}}{\text{先进高速部件完成该功能的时间}}$$

而采用了高速部件后整机性能提高比，即

$$\text{Speedup}=\frac{\text{老计算机完成整个任务的时间}}{\text{采用高速部件的新计算机完成同样任务的时间}}=\frac{1}{1-F+\frac{F}{S}}$$

[例 1-4]　采用新器件使某一功能性能提高 10 倍，但该功能的使用只占原程序运行时间的 40%。请计算新计算机性能改善了多少？

解：

$$F=0.4$$

$$S=10$$

$$\text{Speedup}=\frac{1}{1-0.4+\frac{0.4}{10}}=1.56\text{ 倍}$$

使用 Amdahl 定律要弄清的一点就是 F 值。它是指使用改进提高性能的部件或新增功能部件的那部分任务在**原计算机**中运行占整个任务运行的时间百分比，而不是在改进后的新计算机上实测到的高性能部件或新增功能部件运行的时间百分比(参考习题 1.3)。同时我们从