

内 容 简 介

RISC(精简指令系统计算机)技术是计算机体系结构发展史上的一个重要里程碑。本书全面介绍了RISC的技术,注重阐明RISC的基本思想及原理,并介绍了多种现代RISC产品及其特点。

全书分为两篇。第一篇(第一~第五章)为单发射结构RISC,阐明了RISC技术的基本原理,并介绍了早期RISC的设计过程和物理实现方法。第二篇(第六~第十五章)为多发射结构RISC,重点论述了多发射体系结构的新原理,通过介绍一系列最新的RISC产品,讨论了超标量结构、超流水线结构以及超标量超流水线结构等新型计算机体系结构的技术和特点。

本书可供高等院校计算机专业的高年级学生和研究生阅读,也可供从事计算机技术研究、设计、开发和应用的科技人员学习和参考。

(京)新登字 158 号

RISC

单发射与多发射体系结构

李三立 李亚民 著

☆

清华大学出版社出版

北京 清华园

密云胶印厂印刷

新华书店总店科技发行所发行

☆

开本: 787×1092 1/16 印张: 22.75 字数: 538 千字

1993年12月第1版 1993年12月第1次印刷

印数: 0001—5000

ISBN 7-302-01383-7/TP·531

定价: 16.90 元

序 言

精简指令系统计算机 RISC (Reduced Instruction Set Computer) 的问世是计算机体系结构发展史上的一个里程碑。过去,单处理器主要是靠改进 VLSI 工艺,进而提高工作频率来提高处理器速度,其进展过程较为缓慢。RISC 设计思想强调改进经常使用的基本指令的执行效率,充分利用 VLSI 芯片上执行较快而且控制较简单的寄存器-寄存器操作。此外,RISC 设计思想还十分重视依靠硬件与软件的优化结合,共同提高计算机的速度。这些设计思想为大幅度提高处理器的性能开辟了广阔的前景。在不到 10 年内,RISC 技术使 VLSI 处理器的工作速度从 2~3 MIPS 上升到 200~300MIPS,即已达到了过去超级计算机的 CPU 所能达到的高速度。这一成就不仅从根本上改变了微处理器包含的意义和应用的范围,而且对整个计算机产业结构产生了巨大而深刻的影响。

在近十年内,RISC 本身的定义和研究方向也发生了很大变化。从初期的 RISC 的简化指令系统的思想,发展到现代的 RISC 研究方向,即着重于增加每个周期所能执行的平均指令数。换句话说,RISC 从早期的单发射体系结构已经发展到现代的多发射体系结构。同时,编译优化技术在现代 RISC 系统设计中的地位,也愈来愈重要,它有力地支持 RISC 体系结构潜在性能的发挥。现在,RISC 技术已日趋成熟,而且使用范围也日益扩大,时至今日,几乎所有新设计的微处理器和工作站/服务器都已采用了 RISC 设计技术。

RISC 是一项起步不久的新技术,这就给我们以机遇,进行 RISC 领域内的研究与开发,使我国在 RISC 领域内缩短与国际水平的差距。本书的出版将有益于广大的科技人员学习与掌握 RISC 技术。

本书比较全面地介绍了 RISC 技术。在讲述 RISC 原理时,注重阐明 RISC 的基本思想及其精华,而不拘泥于 RISC 的定义。通过介绍 UCB 的早期 RISC,使读者了解 RISC 设计的全过程与物理实现方法。在论述现代 RISC 结构的新原理(如 cache 一致性和存储管理等)的同时,介绍了一系列最新 RISC 产品的特点。本书主要内容是作者多年来在清华大学讲课资料的基础上整理而成的,其中也包含了一些我们的研究成果。还有一些内容取材于国际上最新发表的有关资料。

本书分为两篇。第一篇是传统的单发射结构 RISC,包括第一到第五章;第二篇是现代的多发射结构 RISC,包括第六到第十五章。多发射体系结构的重点内容为各种具有代表性的 RISC 产品类型,例如嵌入式 RISC、超标量 RISC、超流水线 RISC 和超标量超流水线 RISC,以及当前速度最快的 RISC——DEC Alpha 和当前影响可能最广泛的 RISC——INTEL Pentium。

本书第四、六、十和十一章由李亚民执笔,其余各章由李三立执笔。全书由李三立审阅。付兴钢博士为第七章提供了很多材料,在此表示感谢。

RISC 技术的发展日新月异,基本上每年都有两三种新 RISC 产品问世,资料目不暇接。国际上目前还没有一本质量较好的书可以参考,而有些产品资料还很不系统与完善,加之作者学识水平有限,书中有错误和不足之处,敬请读者批评指正。

作者

1993 年 6 月

目 录

第一篇 单发射结构 RISC

第一章 微计算机技术发展综述	3
第二章 RISC 设计思想和原理	10
§ 2.1 计算机体系结构设计思想的演变.....	10
§ 2.2 RISC 体系结构的特点和定义	18
§ 2.3 先进微处理器体系结构的发展方向.....	26
第三章 UCB RISC I 与 RISC II 体系结构设计	31
§ 3.1 引言.....	31
§ 3.2 UCB RISC I 结构	31
§ 3.3 UCB RISC I 与 RISC II 的流水线.....	38
§ 3.4 UCB RISC I 指令系统特点	41
§ 3.5 RISC II 的数据通路 with VLSI 实现	47
§ 3.6 RISC II 控制部件	54
第四章 SPARC 体系结构	58
§ 4.1 SPARC 概述	58
§ 4.2 SPARC 编程模型	59
§ 4.3 指令系统.....	68
§ 4.4 信号描述.....	77
§ 4.5 流水线和指令执行时序.....	82
§ 4.6 陷阱处理.....	85
第五章 MIPS 体系结构	88
§ 5.1 S. U. MIPS 结构设计的特点	88
§ 5.2 S. U. MIPS 结构及其实现	90
§ 5.3 S. U. MIPS 的异常事故	94
§ 5.4 S. U. MIPS 的组成和数据通路实现	97
§ 5.5 MIPS R2000/R3000 结构	99
§ 5.6 MIPS R2000/R3000 流水线结构	103
§ 5.7 MIPS R2000/R3000 存储管理	108
§ 5.8 MIPS R3010 FPA 浮点处理器	113

第二篇 多发射结构 RISC

第六章 CACHE、存储管理与总线	119
§ 6.1 cache 概述	119
§ 6.2 SPARC CMU 中的 cache 控制器	127
§ 6.3 SPARC CMU 中的存储管理	132
§ 6.4 MBus 原理	135
第七章 多发射结构微处理器与编译优化指令调度	140
§ 7.1 流水线结构与指令调度	140
§ 7.2 指令调度和指令级并行处理	142
§ 7.3 指令调度的方法与分类	144
第八章 INTEL 80860——超级计算超标量 RISC	156
§ 8.1 80860 体系结构特点	156
§ 8.2 80860 的组成	157
§ 8.3 80860 的寄存器堆	164
§ 8.4 80860 的流水线	172
§ 8.5 80860 的指令系统	176
§ 8.6 80860 的存储管理	179
§ 8.7 80860XP 的 cache 管理和多处理功能支持措施	182
§ 8.8 80860XP 的硬件实现	191
第九章 嵌入式超标量 RISC——INTEL 80960 CA/CF	199
§ 9.1 概述	199
§ 9.2 80960 CA/CF 主要设计特点	201
§ 9.3 80960 的组织结构	203
§ 9.4 80960 CA 指令系统	207
§ 9.5 80960 CA 的调用和返回机制	214
§ 9.6 80960 系统控制机构	217
第十章 IBM RS/6000 结构及组成	219
§ 10.1 RS/6000 概述	219
§ 10.2 RS/6000 体系结构	220
§ 10.3 RS/6000 硬件组成原理	225
第十一章 MOTOROLA 88110 超标量 RISC	233
§ 11.1 MC88110 超标量处理器概述	233
§ 11.2 指令系统体系结构	237
§ 11.3 指令的读取与发射	240
§ 11.4 MC88110 执行单元	246
§ 11.5 地址转换机制	249

§ 11.6	cache 操作	251
§ 11.7	外部总线接口	255
§ 11.8	两级 cache 结构	257
第十二章	超流水线 RISC 体系结构 MIPS R4000	258
§ 12.1	MIPS R4000 组成部件	258
§ 12.2	R4000 CPU 流水线结构	262
§ 12.3	R4000 CPU 寄存器和指令系统	268
§ 12.4	R4000 存储管理和系统软件模型	274
§ 12.5	R4000 cache 一致性和多机系统支持	278
§ 12.6	R4000 的引线信号与功能	294
第十三章	超标量超流水线 RISC 体系结构 DEC Alpha	300
§ 13.1	Alpha 21064 的结构	301
§ 13.2	流水线结构	308
§ 13.3	DEC Alpha 指令系统	309
§ 13.4	Alpha 21064 引线功能和外部接口	316
第十四章	超标量 RISC——Super SPARC 体系结构特点	331
§ 14.1	Super SPARC 结构中的流水线	331
§ 14.2	指令发射策略	334
§ 14.3	转移策略	335
§ 14.4	浮点部件	336
§ 14.5	存储器的层次结构	338
§ 14.6	总线操作	340
§ 14.7	工艺实现	341
§ 14.8	第二级 cache 控制器	341
第十五章	双流水线超标量 RISC 体系结构 INTEL Pentium	343
§ 15.1	Pentium 的结构框图及其特点	343
§ 15.2	Pentium 的流水线和指令执行顺序	345
§ 15.3	指令配对法则和转移预测	347
§ 15.4	浮点部件 FPU	348
§ 15.5	片上 cache 与 TLB	349
§ 15.6	多机系统中 cache 的一致性	350
参考文献	355

第一篇 单发射结构 RISC

9410116

第一章 微计算机技术发展综述

在讨论微计算机技术的新进展时,应该说明,现在微计算机与工作站、小型机以至大型机之间的界线已愈来愈模糊了。微处理器芯片的速度已经达到甚至超过了10年前一般大型机的中央处理器的速度;超级微机系统已经采用了小型机和大型机的体系结构,其图形功能有时也可达到1兆像素分辨率。然而,本文仍按传统的看法,以单个VLSI芯片为CPU,并以这种CPU组成的系统称为微计算机。下面仅就微计算机的微处理器芯片、微计算机系统的体系结构、微计算机系统的总线以及微计算机的操作系统等几个主要方面,来讨论近年的进展及其未来发展方向。从中可以看到先进的微处理器体系结构RISC的历史沿革和意义。

1. 微处理器芯片

自从80年代后期,RISC芯片从实验室进入工业产品市场以来,微处理器芯片的速度获得极大的提高。当微处理器沿着CISC道路发展的时候,从1979年最快的1MIPS M68000微处理器,到1987年大多数工作站都采用的速度为2MIPS的M68020,花了8年时间,速度只翻了一倍。那时INTEL 80386刚问世,速度也不过(3~4)MIPS。可是,自RISC问世以来,到1992年已经出现了DEC ALPHA为CPU的芯片,其浮点速度可达200MFLOPS。不过5年间,将微处理器芯片速度提高了50倍。据称,到了2000年,微处理器芯片可达到4000MIPS,INTEL现在已经在着手研制2000MIPS芯片。因此可以认为,RISC是20多年来计算机体系结构中一项革命性成就,它产生的影响是深刻而巨大的。由图1.1可见,RISC技术使80年代中后期的微处理器速度飞速提高,目前已经赶上了大型机中央处理器的速度。100MIPS的微型机即将以台式机型式出现在用户面前,它使微机的应用领域更上了一层台阶。意义更深刻的是:由于RISC芯片速度已经赶上大型

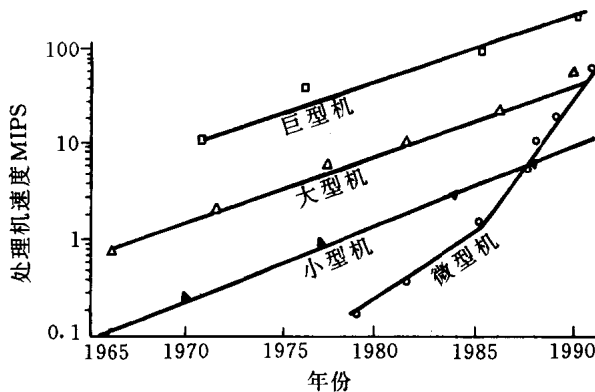


图 1.1 处理机速度增长情况

机,这样就创造了一个条件,即可以把微型机、工作站、小型机和大型机建立在一个二进制兼容的共同平台 RISC/UNIX 之上,依靠不同的技术,实现不同的机型和用途。例如,加强图形功能,可成为工作站;采用双机或多机系统和扩大内存,并增强 I/O 通道能力和外存容量,即可发展成小型机。如再采用并行处理技术,即可达到目前大型机的功能。这样将使微、小、大、巨型机的界线不象传统观念那么清楚,同时会刺激计算机体系结构设计技术,如并行处理技术和图形技术向更精细和更高效能的方向提高。同时,二进制兼容的平台使软件开发更为有效,而且使微、小、大、巨型机连网和共享资源更为方便。

在 RISC 芯片不断发展的同时,CISC 芯片也取得长足的进步。50MHz 的 80486 已经投入生产,其速度也可达到 20MIPS。当然,它所要求的片上晶体管数相对较多。MOTOROLA 的 $0.8\mu\text{m}$ 工艺的 68040 可达 40MHz 工作频率,晶体管数为 120 万个; $0.65\mu\text{m}$ 工艺的 M68050 即将推出,其性能为 68040 的二倍, $0.5\mu\text{m}$ 工艺微处理机芯片也在试制中。

CISC 芯片之所以还在不断发展,是由于要保持软件上的兼容性。可以认为:80 年代微机的不断发展使计算机成为一个大规模生产的产业,这个规模经济产业(硬件和软件产业)的基础是建立一个标准化的平台。这个平台就是 80x86/MSDOS,该平台上的规模性经济可以用以下数据(1991 年数据)来说明:

80x86 的 PC 机生产达 7000 万台

MSDOS 操作系统达 7000 万份

80x86 系统投资达 3400 亿美元

MSDOS 上应用软件达 2 亿件

可以相信,具有如此巨大经济实力支持的 80x86/MSDOS 在未来的年代中还会具有很强大的生命力。CISC 还会在发展中采用 RISC 思想,技术将要不断更新。操作系统 MSDOS 会向 Window 方向发展,CISC 80x86 会逐渐靠近 RISC 的体系结构。大家知道,精简指令系统计算机 RISC 的设计概念的精华不在于减少其指令系统中的指令数目,而是在于减少平均每条指令执行所需的周期数 CPI 值。从 8088/8087 到 80386/80387,再发展到 80486,其中每条指令执行所需的周期数也在不断减少。现在,INTEL 的 80586(现已正式定名为 Pentium)已经采用了第三代 RISC 体系结构中的超标量结构,即处理机中有多个执行部件,其指令执行并行度更高了。

可以看出,从体系结构概念来看,CISC 与 RISC 是殊途同归,都在向减少 CPI 值、向超标量与超流水线结构方向发展。然而,RISC 具有结构简单的优点,因此步伐更快一些,速度提高更快一些。INTEL 的 RISC 芯片 80860 和 80960 都已采用了超标量结构。前者用于通用机,后者适于嵌入式用途。80960 速度可达 60MIPS,但每片只有 20 美元,其性能价格比是相当高的。MOTOROLA 88100/88110、新型的 SPARC 以及 IBM 的 POWER 6000 都采用了超标量结构。POWER 6000 除了采用分开的定点运算和浮点运算多个执行部件结构外,还设置了一个转移处理部件,它把编译优化中的指令调度的部分工作用硬件来实现,做到动态的指令优化调度,使指令执行的并行度大为提高。各方资料表明,IBM 将把 POWER 6000 作为它研究与开发工作的重点。各方报道表明,POWER 6000 处理系列将在 IBM 未来各档产品系统中起到极其重要的作用。

目前采用超流水线结构方案的 RISC 芯片只有 MIPS 公司的 R4000。它共有四级流水线,在提高芯片主频率的情况下,每级流水线节拍又可细分为两个小级节拍,每隔一个小级节拍取出一条指令。如果每个主周期细分的小节拍数愈多,则超流水线结构的 CPI 值愈小,但实际上的工作频率相应地也愈高。因此,可以认为超标量结构与超流水线结构为提高指令执行并行度的概念有所不同。前者是靠空间来换取时间,后者是以时间来换取空间。这两个方向的结构合并,可以形成超流水线的超标量结构。

计算机界普遍关心的问题是:VLSI 微处理器的速度提高有没有极限?随之而来的问题是:VLSI 工艺线宽会减少到什么程度?单片芯片上的晶体管数会增加到什么程度?制造快速处理器的技术发展趋势是什么?目前,象 Pentium 这种高速 RISC 的 VLSI 工艺是采用 $(0.7\sim 0.8)\mu\text{m}$ 工艺,晶体管集成度(单片)可达 300 万个,工作主频为 50MHz。HP 公司即将推出的 (PA)RISC 将用 CMOS 工艺做到 90MHz(五级流水线),这已经和 ECL 工艺相当。但是,要进一步提高单片上晶体管集成度,就要增加芯片面积,这将引起成品率下降,并使价格随之迅速上升。现在国外快速微处理器制造技术的一个重要趋势是采用低电压逻辑和多芯片模块(MCM)。降低电源电压可以减少电流对寄生电容的充电时间,减少逻辑电路延迟。如 IBM RS/6000 41MHz 的 550 型中采用的芯片,采用 $0.5\mu\text{m}$ 工艺,3.6V 逻辑。有的公司芯片内部逻辑用 5V 电压,但芯片与外界接口的输入/输出信号采用 1V 电平,因为这些信号线上的寄生电容比芯片内部大得多。MCM 指的是把多个单芯片连接起来组成处理器,但这些多个单芯片是封装在一个组件中。如 IBM 的 POWER 6000 RISC 处理器共有 700 万晶体管,它分成 9 个单芯片(其中 450 万晶体管是用于 cache 的),这样单芯片的集成度不必太高,从而可降低价格。但随之而产生的问题是芯片之间引线的张力、湿度问题和热敏问题。此外,芯片之间的引线的寄生电容和电感都要设法减少。引线长度减少 6 倍,寄生电感可减少 60 倍,寄生电容可减少 6 倍,临界数据路径的延迟可减少 3 倍。目前,提高单片芯片本身集成度仍是发展趋势。亚微米的工艺还在向 $0.5\mu\text{m}$ 过渡。INTEL 计划做 5000 万到 1 亿个晶体管的 CPU,它的 Micro 2000 体系结构包括 2000 MIPS 的微处理器,并与 80386 以及 MSDOS/OS 2/UNIX 结构二进制兼容。其它工艺如 ECL 和 BiMOS 也在进展中。但总的来说,CMOS 工艺比较适合于晶体管数较多的多个执行部件的超标量结构,而 ECL 与 BiMOS 工艺比较适合于要求主频较高但晶体管数较少的超流水线结构。

2. 微机系统体系结构

以上所述的微处理器(无论是 CISC 还是 RISC)速度的迅速提高,使它与存储器 DRAM 的速度差距愈拉愈大(见图 1.2)。这一趋势对微机体系结构产生深刻影响,使微机系统体系结构中的 cache 显得更为重要。解决问题的办法是在 CPU 芯片上添设片上 cache。但如上所述,片上 cache 容量增大,芯片面积将增加很多,从而导致成品率下降和成本上升,因此目前片上 cache 容量最多在 8K 字节左右,这显然是不够的。因此,在高档微机系统中都必须增添片外的大容量 cache,如 64K,128K 以至 256K 字节。这种片上和片外多层次 cache 是目前微机体系结构中通常遇到的问题。此外,高速 CPU 与高速 cache 的配合,又要求有高速的总线来传输数据,总线的频宽要求为 160MB/s,甚至更高。这是

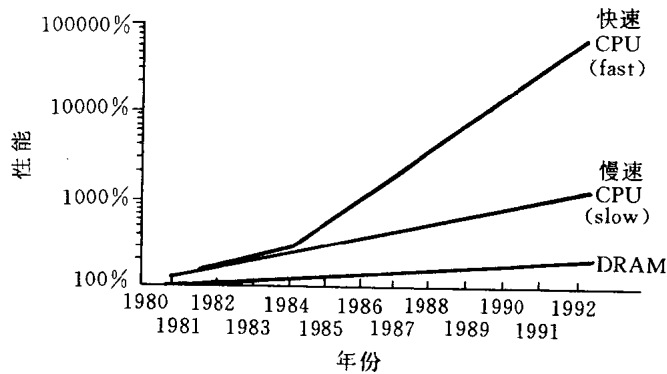


图 1.2 CPU 与 DRAM 的速度差距

目前标准的系统总线所远远不能满足的。因此现代高档微机系统都另设内部总线，这内部总线有时具有两个层次，最高速度的内部总线提供高速 CPU 与 cache 之间传输通路，次高速度的内部总线为存储管理部件与大容量 DRAM 提供传输通路。另外有两种与外围设备相连的外部总线，一种较高速的外部总线与图形卡、DMA、局部网络卡以及 SCSI 相接，另一种 8 位的较低速的外部总线联接软磁盘、键盘、串行接口、ISDN 以及音频端口。这种多层次总线结构见图 1.3。这一结构随着 CPU 速度继续提高以及多媒体环境的外围设备多样化而显得更加重要。

近年来很多有名的微机公司推出多机微机系统，一般采用四个到八个微处理器，使用各种版本的多机 UNIX 操作系统。整个系统的功能可以达到 50MIPS，甚至 100MIPS 以上，其应用领域已经渗透到传统超级小型机的领域。这种系统的体系结构如图 1.4 所示。

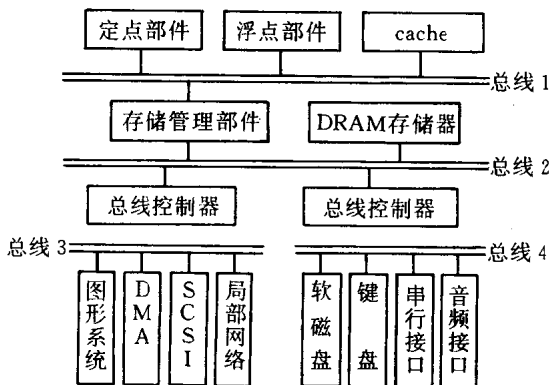
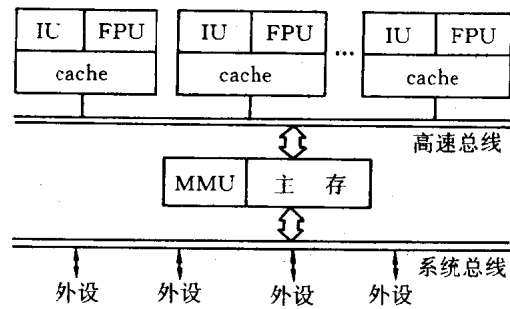


图 1.3 多层次总线体系结构



IU: 整数部件 FPU: 浮点部件
cache: 高速缓存 MMU: 存储管理

图 1.4 多机微机系统结构框图

多机微机系统的发展潜力很大。目前一般采用 80386 CPU 或 80486 CPU，处理器和 cache 做在一块板上，cache 容量为 64KB 或 128KB。如果把 CPU 改成 (40~50)MIPS 的 RISC，那么这个多机系统的功能将超过一般大型机。而且这种系统的扩展性很好，如果增

加其中的 CPU 板,选择高性能的高速总线,则系统的性能还可以升级。因此,这种系统结构很值得我国计算机界重视。其中比较关键的技术是 cache 的一致性、多机系统的处理器之间通讯协议、高速总线的结构以及多机 UNIX 系统。

3. 总线

如前所述,微机 CPU 速度的迅速提高,使高速总线和层次式总线结构设计的重要性更加突出了,这些总线结构也是国际上各大计算机公司正在努力解决的问题。过去的 AT 总线、MULTIBUS I、VME 总线、以至 EISA 与 MCA 总线的速度都已经不够了。其出路是设计新型号总线或者另外设计特定的总线。

目前国际上已采用或正在推出的总线标准及其应用范围见图 1.5。其中 AT 总线、EISA、MCA 低档型号主要用于微机,MULTIBUS I 与 II 主要用于工业控制用微机,VME 则多用于工作站。EISA、MULTIBUS I 和 VME 64 也可用在多机微机系统。高档 MCA 型号和 VME 等总线将用于未来的更高速的微机系统。

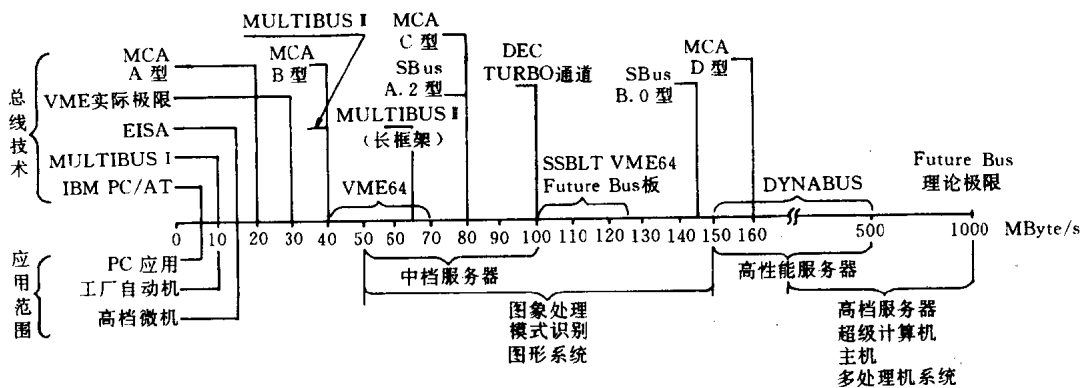


图 1.5 系统总线技术与应用范围

总线设计的新技术要考虑到 cache 一致性和信息传递协议,目前的新总线要更有效地支持多机系统,因此在这两方面做了精细的改进。此外,为了加速数据传送,除了采用成块传送以外,还有突发传送(burst transfer)功能。另外高速总线还要考虑到传输延迟、接地、干扰屏蔽等很多工程性问题。总之,在未来新微机系统中,总线设计已成为极其重要的问题,总线标准也是世界各大计算机公司激烈竞争的一个焦点。谁能掌握处理器标准、总线标准和操作系统标准,谁就能在未来 10 年中微机、工作站、小型机以至大型机各领域占统治地位。

4. 操作系统

新一代的微机操作系统要考虑的关键问题包括:

- 支持 Window 功能
- 支持多线索调度
- 支持多处理机功能

- 支持面向对象技术

未来的操作系统则将对多媒体技术提供全面支持。

下面介绍 1991 年与 1992 年推出的 32 位微机中采用的几种操作系统及性能。

(1) Microsoft 的 Windows NT (New Technology)

- 多线索
- 多任务
- 对称式多处理机
- 支持多种应用程序接口 API
- 支持多文件系统

(2) IBM 的 OS/2 2.0*

- 多线索
- 多任务
- 支持多文件系统
- 支持多种应用程序接口 API
- 表示管理功能

(3) SOLARIS (SUN 与 INTEL 联合开发)。

- 开放式窗口 GUI
- 开放式网络计算
- 面向对象的开发工具
- 支持对称式多处理机
- 多线索

(4) Mac System 7.0 (APPLE 与 IBM 联合开发)。

- 数据访问管理器
- 印刷和页面描述
- 进程之间通讯
- 多任务
- 虚拟存储

微机操作系统中的一个重要趋势是面向多厂家,即可容纳不同的操作系统的应用软件,可支持微机,也可支持工作站。此类操作系统称为跨平台(Cross Platform)操作系统。APPLE 正在开发的 PINK 操作系统是一个突出的典型。它可以在 MACINTOSH 上运行,也可以在其它机型上运行。它可以运行 MAC OS System 7.0,也可运行 OS/2 和 UNIX 软件,还可在 IBM 的 RS6000 上运行,并支持多媒体功能。据称,这个 PINK 操作系统是促使 IBM 高级总裁做出与 APPLE 联盟决定的最有力的因素,它将大大加强 IBM 在软件方面的力量。

其它 Microsoft 的 NT 操作系统也在积极开发多媒体功能。

综上所述,从宏观来看,80 年代微计算机的发展,对于计算机事业以至全世界人类生活做出的最大贡献是建立了一个全世界都接受的标准化平台 80x86/MSDOS,从而使微计算机成为一项大规模经济的产业,使计算机应用深入到各个领域。

80年代后期微机史上发生了深刻的变化,即 RISC 诞生了。它对计算机硬件与软件技术以至对于计算机产业结构都会产生深刻的影响。因此,各大公司都在努力使自己的 RISC 产品成为 90 年代的标准。目前计算机界动荡、分化和重新结盟,其目的就是追求一个新的标准平台 RISC/UNIX。这个平台上的投资和影响将比 80 年代的平台 80x86/MS-DOS 更大。可以肯定,未来的处理机芯片都将向 RISC 设计概念靠拢和发展;未来的操作系统也向 UNIX 概念靠拢和发展。

第二章 RISC 设计思想和原理

§ 2.1 计算机体系结构设计思想的演变

在 50 年代,计算机设计都是逐个进行的,每个计算机有它自己的指令系统,用户按照其汇编符号编写程序,当时还用到了—些程序设计自动化的技术。但是,那时还没有计算机产品系列化和软件兼容性的概念。

60 年代以后,程序设计自动化技术获得了发展,高级语言(如 FORTRAN、ALGOL)逐渐成熟。由于应用的需要,软件编写的工作量愈来愈大,这就要求用户花很多功夫编写好的软件以便在同一系列但不同档次上的计算机中都可运行。减轻软件编写工作量对计算机设计提出了新的要求。

1964 年 IBM 推出的 IBM SYSTEM 360 标志着计算机体系结构发展史上的一个重要里程碑。IBM SYSTEM 360 首次在计算机体系结构与计算机组织之间划出一道清楚的分界线:计算机体系结构是机器语言程序设计员为编写程序所必须看到的一个计算机的抽象结构。计算机组织是为实现该结构的硬件组成。

因此,对于某一个产品系列,其计算机体系结构可以是相同的,系列中每一个档次的产品的计算机组织却可以有差别。但在系列产品中,各个档次的产品在软件上具有兼容性。这样,对于不同要求的用户,可以提供不同层次、不同价格的计算机产品,但用户所用的软件却可以向上兼容,而且用户可以花较少的费用使自己的计算机向上升级,同时可保持用户投资的软件继续有效。当时,这一概念十分重要,它打破了 50 年代计算机的“手工业式”生产方式,使计算机硬件和软件都获得迅速发展,奠定了计算机发展的基础。这一概念至今仍有不可动摇的地位,仍是计算机体系结构设计者必须遵循的依据。

2.1.1 传统的计算机体系结构设计技术

60 年代中期,微程序是实现计算机系列产品的体系结构,并保证软件向上兼容的一项重要技术。在系列产品中,要求该系统中的计算机的指令系统有一个核,这个核是不变的,才能保持软件兼容。此外,在系列产品中,从低档产品到高档产品,性能逐步提高,往往要求指令系统也增加功能更强的一些指令,当然计算机组织的复杂程度也随之增加,价格也随之而增加,但这样可满足一些高要求的用户。当时,微程序技术是满足上述体系结构设计要求的有效而可行的技术。在微程序设计中,相应于上述的一个指令系统的核,微程序存储器的核心也是保持不变的,要扩展一些功能较强的指令,只要逐渐扩充微程序的存储器就可以了。因此,自 60 年代以来,几乎所有的大型机、小型机和微型机的控制部件都采用了微程序技术设计其控制指令执行的部件。

应该指出的是,微程序技术之所以风行,还有一个计算机工艺技术的背景。在 60 到 70 年代,计算机的主存储一直采用磁心体,而快速的微程序存储则采用半固定存储器,70

年代则开始采用小型半导体存储器。当时的 CPU 已采用很高速度的双极型半导体逻辑，它比磁心主存快(5~10)倍，而与小型快速微存储却相当。当时传统的计算机的机器周期

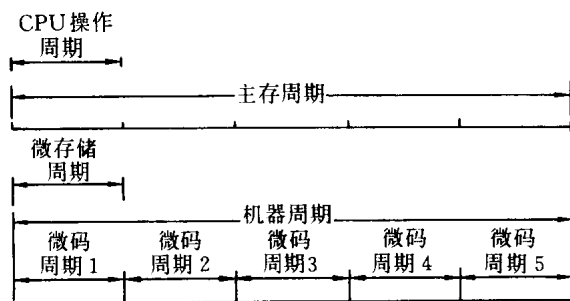


图 2.1 机器周期与微码周期的匹配

一般相当于主存的周期，它正好等于(5~10)个微存储周期(见图 2.1)。这与微程序技术中，一条指令是由(5~10)个微指令操作组成的情况是很匹配的。可见，微程序控制技术的发展是受到当时的计算机工艺技术支持的。

此外，在 60 年代和 70 年代，计算机体系结构设计还受到一些传统思想的深刻影响，这些思想对于指令系统的选择起了很大的作用。这些传统思

想认为指令系统愈丰富愈好，其理由是：

1. 指令系统愈丰富则功能愈强，而且编译程序愈好写。尤其是要增加存储器操作的指令。当时认为：在寄存器-寄存器操作指令、存储器-寄存器操作指令以及存储器-存储器操作指令等三种指令中，以第三种指令的效率最高。这种思想，在 M68000 微处理器设计中表现尤为突出，M68000 采用了大量存储器-存储器操作指令。

我们用十分简单的例子说明：

$$B+C \rightarrow A$$

如采用寄存器-寄存器操作，要用四条指令才能完成，即

```
LOAD  B,rB
LOAD  C,rC
ADD   rB+rC→A
STORE rA,A
```

如采用存储器-寄存器操作，则需用三条指令：

```
LOAD  B
ADD   C
STORE A
```

但如采用存储器-存储器操作，则用一行指令就可完成，即

```
ADD   B+C→A
```

其中 A、B 和 C 是存储器单元。

2. 指令系统愈丰富，愈可减轻软件危机。由于软件成本上升而硬件成本下降，计算机设计者想尽可能地把计算机执行高级语言的功能移到硬件上去，使机器语言很象高级语言的语句。这样可以缩小程序设计语言与机器语言之间的所谓“沟距”，从而使程序代码缩短，减少软件上耗费的愈来愈大的费用。

3. 指令系统丰富，尤其是操作步骤较多的存储器操作指令增多，可以改善体系结构的质量。60 至 70 年代，衡量计算机体系结构质量的重要标准是“代码长度”与“存储效率”。当时存储器还采用磁心，存储容量较小，而半导体存储器也刚开始采用，价格也较贵。

程序愈小,则存储效率愈高。

代码长度是指一段程序中指令和数据总共占用的存储总容量(以位“b”计)。图 2.2 表示三种类型操作指令所占用的存储总容量 $M, M=I+D$ 。其中 I 是指令占用位数, D 是数据占用位数。假定操作码为 8 位,地址码是 16 位,寄存器共 $2^4=16$ 个。执行 $B+C \rightarrow A$, 数据字长 32 位。这三种不同操作指令中,以存储器-存储器操作指令的存储效率最高。

(1) 寄存器-寄存器操作指令

LOAD	rB	B	
LOAD	rC	C	
ADD	rA	rB	rC
STORE	rA	A	

```
LOAD B,rB
LOAD C,rC
ADD rB+rC→rA
STORE rA,A
I=104 b
D=96 b
M=200 b
```

(2) 存储器-寄存器操作指令

LOAD	B
ADD	C
STORE	A

```
LOAD B
ADD C
STORE A
I=72 b
D=96 b
M=168 b
```

(3) 存储器-存储器操作指令

ADD	B	C	A

```
ADD B+C→A
I=56 b, D=96 b, M=152 b
```

图 2.2 三种不同操作指令

当时很多研究工作者都认为应当大量采用存储器-存储器操作指令,指令系统中含有这种指令愈多,则程序愈紧凑,存储效率愈高。有的作者甚至认为不能再使用寄存器-寄存器操作指令,但实际上存储器操作是很复杂的。

4. 指令系统的复杂便于追求软件兼容。如以 INTEL 的 80x86 系列为例。8086 设计要考虑到 8085 原有的几个 8 位寄存器的指令,80286 则要考虑到与 8086 兼容的指令。如此发展,则在不断扩充其性能的同时,还要考虑与前代处理器的兼容性,其指令系统必然愈来愈庞大了。

归纳以上的情况,60 年代末期和 70 年代以至 80 年代初的计算机体系结构设计中有两个特点:

1. 把存储效率作为体系结构设计质量的重要衡量手段,大量采用存储器-存储器操作指令,甚至采用面向存储器堆栈操作的体系结构。为了充分利用内存,还采用了可变字