

955
7926 49
747

面向 21 世纪计算机专业本科系列教材

计算机系统结构

主编 尹朝庆
编著 尹朝庆 唐建雄 陈莉



A0935493

华中理工大学出版社

图书在版编目(CIP)数据

计算机系统结构/尹朝庆 主编
武汉:华中理工大学出版社, 2000年3月
ISBN 7-5609-2142-6

I. 计…
II. ①尹… ②唐… ③陈…
III. 系统结构-高等学校-教材
IV. TP303

计算机系统结构

主编 尹朝庆

责任编辑:谢燕群
责任校对:戴文遐

封面设计:刘 卉
责任监印:张正林

出版发行:华中理工大学出版社
武昌喻家山 邮编:430074 电话:(027)87542624

经销:新华书店湖北发行所

录排:华中理工大学出版社照排室
印刷:湖北省新华印刷厂

开本:787×960 1/16	印张:22.25	字数:360 000
版次:2000年3月第1版	印次:2000年3月第1次印刷	印数:1—5 000
ISBN 7-5609-2142-6/TP·366		定价:25.80元

(本书若有印装质量问题,请向出版社发行部调换)

前 言

随着现代科学技术的飞速发展,计算机应用领域对计算机性能的要求越来越高。高性能计算机也一直是计算机科学技术的主要发展方向之一。20世纪的最后10年,计算机系统结构在许多方面有了重大进展,例如,超标量、超流水线和超标量超流水线相结合的系统结构在微处理机中已经得到广泛应用;多处理机系统及其互连网络有了进一步的提高和完善,体系结构日趋成熟;大规模并行处理系统的典型产品相继问世,已成为计算机家族中的佼佼者;由高性能工作站通过局域网互连构成的机群系统因可扩展性好等优点而倍受青睐。高性能计算机是十分活跃的生产力,在信息化社会中占有极其重要的地位,是衡量国家综合国力的重要标志。高性能计算机的推广应用正在敲响世纪之交的大门,将不断向纵深发展,其应用领域将进一步不断扩大。

计算机系统结构由低向高级发展的过程,也就是并行处理技术不断发展的过程。在计算机技术高速发展的今天,只要涉及到计算机系统结构的有关知识,就离不开并行处理的概念和技术。近20年来,有关计算机系统结构与并行处理的新技术不断涌现,为了更加系统全面地介绍计算机系统结构,及时补充一些新的内容,我们编著了这本书。

本书作者长期从事计算机系统结构等系统构成技术有关课程的教学和科研工作,在查阅和综合分析了大量有关资料的基础上编写完成了本书。全部内容分为两大部分,第一部分是构成高性能计算机系统的基本概念和有关技术及其基本模块,包括RISC、流水线技术、向量处理技术、存储系统和互连网络等。第二部分是SIMD计算机、多处理机系统与机群系统、数据流计算机的结构与性能分析。

本书力求内容新颖全面、重点突出,语言精炼易懂。各章都附有习题。可作为高等院校计算机及相关专业的高年级本科生、研究生的教材,也可作为有关专业的教师和科技工作者的参考书。

本书作为计算机系统结构的教材,第一部分的第1~4章应是课程的必讲部分,第二部分的第5~7章可根据需要选学部分内容。

本书的第 1、2、5、6、7 章和第 3 章的 3.3 节、第 4 章的 4.3 节由尹朝庆编写,第 3 章的 3.1、3.2、3.4 和 3.5 节由陈莉编写,第 4 章的 4.1、4.2 和 4.4 节由唐建雄编写。尹朝庆任主编,负责审定全书的内容。

作者十分感谢“面向 21 世纪计算机专业本科系列教材”编委会对本书的指导和支持。

由于作者的水平和经验有限,难免有疏漏及不当之处,希望读者批评指正。

作者

1999 年 8 月



计算机系统结构导论

计算机系统性能的不不断提高,主要是依靠器件的不断更新和计算机系统结构的发展。本章介绍计算机系统结构的基本概念和计算机系统并行性发展的技术途径与分类;讨论系统结构的一般性能评价标准;研究机器指令格式的优化设计方法和精简指令系统计算机的特点及关键技术。

1.1 计算机系统结构的基本概念

回顾计算机的发展历史,可以看出,计算机系统性能的不不断提高主要靠器件的变革和计算机系统结构的改进。自第一台电子计算机问世以来,以器件的发展为标志,计算机经历了电子管、晶体管、中小规模集成电路和大规模超大规模集成电路4代。由于器件的迅速发展,使得计算机硬件在体积、重量、速度、可靠性、稳定性等性能上有了极大的改善,而且价格不断降低。可以说,器件技术的突飞猛进,为计算机的发展提供了必不可少的物质基础,它是推动计算机发展的重要因素之一。器件的换代是计算机换代的突出标志。

但是,还应该看到,仅有器件的发展是远远不够的。人们还必须研究如何最合理地组织这些器件,如何最大限度地发挥这些器件的作用,如何构成综合性能最佳的系统,这就是计算机系统结构要研究的问题。恩斯洛(P. H. Enslow)曾经比较了1965年至1975年的器件延迟时间和计算机指令执行时间之间的关系。结果表明,这10年间,器件的更新使器件延迟时间降低至原来的十分之一,但计算机指令执行时间却降低至原来的百分之一。由此可见,在这10年中,计算机性能提高的幅度比器件性能提高的幅度大得多。这种情况在近几年的计算机发展中更为明显。因此,除器件的发展以外,计算机系统结构的发展也是推动计算机发展的一个重要因素。

第一台电子计算机问世以来,计算机已经历了5次更新换代。第一代计算机(1945~1954)将电子管和继电器存储器用绝缘导线互连在一起。由单个

CPU 构成, CPU 用程序计数器和累加器顺序完成定点运算。采用机器语言或汇编语言, 用 CPU 程序控制 I/O。第二代计算机(1955 ~ 1964)采用分立式晶体管和铁氧体磁芯存储器, 用印刷电路将它们互连起来。采用了变址寄存器、浮点运算、多路存储器和 I/O 处理机。采用有编译程序的高级语言、子程序库、批处理监控程序。第三代计算机(1965 ~ 1974)采用小规模或中规模集成电路和多层印刷电路。微程序控制在这一代开始普及。采用了流水线、高速缓存和先行处理技术。软件方面采用多道程序设计和分时操作系统。代表性的系统有 IBM360/370 系列、CDC6600/7600 系列等。第四代计算机(1974 ~ 1991)采用大规模或超大规模集成电路和半导体存储器。出现了用共享存储器、分布存储器或向量硬件构成的不同结构的并行计算机, 开发了用于并行处理的多处理机操作系统、专用语言和编译器, 以及用于并行处理或分布处理的软件工具和环境。代表性的系统有 VAX9000、CrayX-MP、IBM3090VF 等。第五代计算机(1991 ~ 现在)采用 VLSI 工艺, 具有更完善的高密度、高速度的处理机和存储器芯片。它的最重要特点是进行大规模并行处理, 采用可扩展和容许时延的系统结构。代表性的系统有 Fujitsu 的 VPP500、Cray Research 的 MPP、Thinking Machines 公司的 CM-5、Intel 超级计算机系统 Paragon、SGI 的 Origin2000 和 Sun 公司的 10000 服务器。由此可以看到, 计算机换代的标志主要有两个, 第一是计算机的器件, 第二是计算机系统的结构。

推动计算机系统结构发展的关键技术是计算机并行处理技术。计算机系统结构从低级向高级发展的过程也就是并行处理技术不断发展的过程。最早的计算机并行是位并行方式。1953 年完成了具有位并行运算的第一台商业计算机 IBM701。同一时期也出现了 CPU 与 I/O 设备在一定程度上的并行。到 60 年代, 并行性进一步发展, 研制出了流水线单处理机系统, 如 1964 年完成的 CDC6600, 设有乘、除、加、移位、布尔运算等操作的 10 个独立的功能部件, 它们可以被多个外围处理机分时复用。1967 年的 IBM360/91 的机器指令的取指、译码、地址计算等均采用流水线方式重叠地工作。在 1970 年至 1980 年这段时间里, 大规模集成电路的快速发展, 导致生产出向量计算机、阵列计算机等多种多样的并行处理系统。1976 年研制的 Crey-1 就是比较成功的向量流水线计算机, 运算速度可达每秒一亿三千万次浮点运算。80 年代以来, 计算机的体系结构又有了突破性的进展, 最具代表性的是精简指令系统计算机(RISC)、数据流计算机等。这些系统结构的出现, 使得计算机的并行处理技术向更高的方向发展。

1.1.1 计算机系统的层次结构

计算机系统(Computer System)由硬件(Hardware)和软件(Software)组成,按功能划分成多级层次结构,如图 1.1 所示。图中每一级对应一种“机器”,在这里,“机器”只对这一层次的观察者而存在。从某一层次的观察者看来,只是通过该层次的语言来了解和使用机器,不必关心其内层的那些机器是如何工作和如何实现各自功能的。这一概念称为透明性(Transparency)。

层次结构中的第 0 级机器由硬件实现,第 1 级机器由微程序(固件)实现,第 2 级至第 6 级由软件实现。由软件实现的机器称为虚拟机器,以区别由硬件或固件实现的实际机器。

第 0 级和第 1 级是实现机器指定功能的中央控制部分。它根据各种机器指令操作所需要的控制时序,配备一套微指令,编写出微程序,来控制信息在各寄存器之间传送,这就是第 1 级机器。实现这些微指令本身的控制时序只需要很少的逻辑线路,可采用硬联逻辑实现,它就是第 0 级机器,是机器的硬件内核。

第 2 级是传统机器语言机器。这一级的机器语言是该机的指令系统。机器语言程序员用这级指令系统编写的程序由第 1 级的微程序进行解释。

第 3 级是操作系统机器。第一级的机器语言中的多数指令是传统机器的指令,如算术运算、逻辑运算和传送等指令。此外,这一级还提供操作系统级指令,例如,打开文件、读/写文件、关闭文件等指令。用这一级语言编写的程序中的那些与第 2 级指令相同的指令由微程序解释,而其中的操作系统级指令则由操作系统进行解释。操作系统是运行在第 2 级机器上的解释程序。

第 4 级是汇编语言机器。这一级的机器语言是汇编语言。用汇编语言编写的程序首先被翻译成第 3 级或第 2 级语言,然后再由相应的机器进行解释。完成汇编语言程序翻译的程序称为汇编程序。

第 5 级是高级语言机器。这一级的机器语言就是各种高级语言。用高级语言编写的程序一般由编译程序翻译成第 4 级或第 3 级机器上的语言。个别的高级语言也用解释的方法实现。

第 6 级是应用语言机器。这一级的机器语言是应用语言,是面对非计算机专业人员直接使用的语言。只需在用户终端用键盘或其他方式发出服务请求就能进入第 6 级的信息处理系统。

把计算机系统按功能划分成多级层次结构,有利于正确理解计算机系统的工作,明确软件、硬件和固件在计算机系统中的地位和作用;有利于理解各

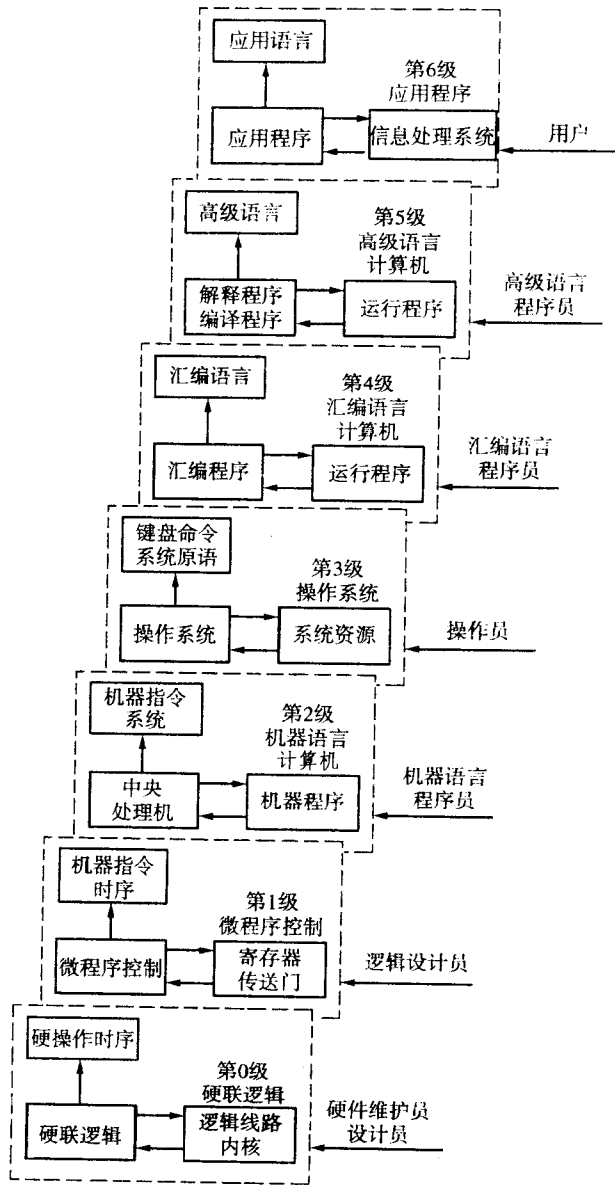


图 1.1 计算机系统层次结构

种语言的实质及其实现;有利于探索虚拟机器新的实现方法,设计新的计算机系统。

可以大致认为第0级至第2级是计算机组织与结构,第3级至第5级属于系统软件,第3级与计算机系统结构密切相关,第6级是应用软件。由于技术的飞速发展,曾经用软件实现的功能已经用硬件或固件来实现,因此,硬中有软、软中有硬、相互转换、彼此渗透,是当今计算机系统中软硬结合的现实。

计算机系统结构主要研究软件、硬件功能分配和确定软件与硬件之间的界面,即哪些功能由软件完成,哪些功能由硬件完成。

1.1.2 计算机系统结构与组成及实现

计算机系统结构(Computer Architecture)也称为计算机体系结构,这个名词从70年代开始被广泛采用。但是,由于计算机软、硬界面在动态变化,至今有各种各样的理解,很难有一个通用的定义。

1964年,G.M. Amdahl在介绍IBM360系统时提出:计算机系统结构是程序员看到的计算机的属性,即程序员为编写出能在机器上正确运行的程序时所必须了解的机器概念性结构和功能特性。然而从计算机系统的层次结构来看,不同级的程序员所看到的计算机属性显然不同。实际上,Amdahl等人提出的计算机系统结构定义中的程序员是指机器语言程序员或编译程序设计者所看到的计算机属性,是硬件系统的概念结构及其功能特性。

对于目前的通用型机器,计算机系统结构一般包括:

- ①数据表示,即硬件能直接识别和处理的数据类型和格式等。
- ②寻址方式,包括最小寻址单位,寻址方式的种类、表示和地址计算等。
- ③寄存器组织,包括操作数寄存器、变址寄存器、控制寄存器及某些专用寄存器的定义、数量和使用约定。
- ④指令系统,包括机器指令的操作类型和格式、指令间的排序方式和控制机构等。
- ⑤存储系统,包括最小编址单位、编址方式、存储容量、最大可编址空间等。
- ⑥中断机构,包括中断的类型、中断分级、中断处理程序的功能和入口地址等。
- ⑦机器工作状态(如管态、目态等)的定义和切换。
- ⑧机器级的I/O结构,包括I/O系统的联结方式、设备的访问方式、处理机/存储器与I/O设备间数据传送的方式和格式、传送的数据量、I/O操作的结

束与出错标志等。

⑨信息保护,包括信息保护方式和硬件对信息保护的支持等。

这些即是程序员为了使其编写的程序能在机器上正确运行,所需要了解和遵循的计算机属性。

计算机组成(Computer Organization)是计算机系统结构的逻辑实现,包括机器级内的数据流和控制流的组成以及逻辑设计等。计算机组成的任务是在计算机系统结构确定分配给硬件系统的功能及其概念结构之后,研究各组成部分的内部构造和相互之间的联系,以实现机器指令级要求的各种功能和性能。这种相互联系包括各功能部件的配置、相互连接和相互作用。各功能部件的性能参数相互匹配,是计算机组成合理的重要标志,因而相应地就有许多计算机组织方法。例如,为了使存储器的容量大、速度快,研究出层次存储系统和虚拟存储技术。在层次存储系统中,又有高速缓存、多体交叉编址存储、多寄存器组和堆栈等技术。为了使输入/输出设备与处理机间的信息流量达到平衡,研究出通道、外围处理机等方式。为了提高处理机速度,研究出先行控制、流水线、多执行部件等方式。在各功能部件的内部结构研究方面,产生了许多组合逻辑、时序逻辑的高效设计方法和结构。例如,在运算器方面,出现了多种自动调度算法和结构等。

计算机组成的设计是按希望达到的性能价格比,最佳、最合理地把各种设备和部件组成计算机,以实现所确定的计算机系统结构。一般计算机组成设计包括数据通路宽度的确定,各种操作对功能部件的共享程度的确定,专用功能部件的确定,功能部件的并行性确定,缓冲器和排队技术的确定,控制机构的设计,可靠性技术的确定等。

计算机实现(Computer Implementation)是计算机组成的物理实现,包括处理机、主机等部件的物理结构,器件的集成度和速度,器件、模块、插件、底板的划分与连接,专用器件的设计,微组装技术,信号传输,电源、冷却及整机装配技术等。它着眼于器件技术和微组装技术,其中,器件技术在实现技术中起着主导作用。

计算机系统结构、计算机组成和计算机实现是3个不同的概念。系统结构是计算机系统的软、硬件的界面;计算机组成是计算机系统结构的逻辑实现;计算机实现是计算机组成的物理实现。它们各自包含不同的内容,但又有紧密的关系。

还应看到系统结构、组成和实现所包含的具体内容是随不同机器而变化的。有些计算机系统中作为系统结构的内容,在其他计算机系统中可能作为组成和实现的内容。开始是作为组成和实现提出来的设计思想,后来可能被

引入系统结构中。例如,高速缓冲存储器一般是作为组成的内容提出来的,其中存放的信息全部由硬件自动管理,对程序员来说是透明的。然而,有的机器为了提高高速缓存的使用效率,设置了高速缓存的管理指令,使程序员能够使用这些指令参与高速缓存的管理。这样,高速缓存对程序员来说就是不透明的,也就是说,高速缓存成为系统结构的内容。

由于计算机组成和计算机实现密切相关,难以明确区分,有人把这两者统称为计算机实现,即包括计算机系统的逻辑实现和物理实现。计算机系统结构设计任务是进行软、硬件功能分配,确定传统机器级的软、硬件界面。但作为“计算机系统结构”这门学科来讲,实际上包括了系统结构和组成两个方面的内容。因此,计算机系统结构是研究软、硬件的功能分配以及如何最佳、最合理地实现分配给硬件的功能。

1.1.3 计算机系统的特性

计算机系统在功能和结构方面都具有明显的多层次的特性,此外,从不同角度看来,计算机系统还具有下述重要特性。

1. 计算机等级

通常把计算机系统分为巨型、大型、中型、小型、微型等若干等级。但是,随着技术进步,各等级的计算机性能指标都不断提高,以致30年前的一台大型机的性能甚至比不上当今一台微型计算机的性能。因此,如果按性能指标来划分计算机等级,那么,一台计算机的等级将随时间而下移。显然,把30年前生产的大型机现在称为微型机是不合适的。各型机器的性能、价格随时间变化的趋势大致可以用图1.2示意说明,其中虚线称为等性能线。由图可见,各型机器所具备的性能随时间是在动态地下移的,但价格却在相当长一段时间内基本不变,因此,有人就主张用价格来划分巨型机到微型机的不同等级。

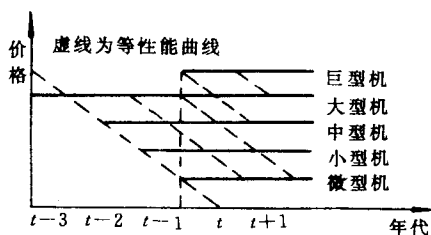


图 1.2 计算机等级与价格、性能关系示意图

计算机等级的发展遵循以下3种不同的设计思想:

① 在本等级范围内以合理的价格获得尽可能好的性能,逐渐向高档机发展,称为最佳性能价格比设计。

②只求保持一定的合用的性能而争取最低价格,称为最低价格设计。其结果往往是从低档向下分化出新的计算机等级。

③以获取最高性能为主要目标而不惜增加价格,称为最高性能设计。其结果是产生当时最高等级的计算机。

第一类设计主要针对大、中型计算机用户的需要,设计生产出性能价格比更好的中型计算机和超小型计算机。第二类设计以普及应用计算机为目标,设计生产数量众多的微、小型计算机。第三类设计只满足少数用户的特殊需要。

从系统结构的观点来看,各型计算机的性能随时间下移,实质上就是在低档(型)机上引用甚至照搬高档(型)机的系统结构和组成。这种低档机承袭高档机系统结构的情况正符合小型机和微型机的设计原则,即充分发挥器件技术的进步,以尽可能低的价格实现高档机已有的那些结构和组成,而不花费很大力量去研究和采用新的系统结构和组成技术。这有利于计算机工业的发展,有利于普及应用计算机。我们已经看到,系统结构和组成下移的速度越来越快,例如,超高速缓冲存储器和虚拟存储器从大型机下移到小型机所花时间不到6年,巨型阵列机问世不过7年,小型机上就有了可扩充的高速阵列处理部件。

2. 计算机系列

系列机概念是指先设计好一种系统结构,然后,软件设计者按这种系统结构设计它的系统软件,硬件设计者根据机器速度、性能、价格的不同,选择不同器件,采用不同的硬件技术和组成与实现技术,研制并提供不同档次的机器。系列机必须保证用户看到的机器属性一致。例如,IBM AS400系列,数据总线有16、32、64位的区别,但数据表示方式一致。系列机必须软件兼容。系列机软件兼容是指同一个软件(目标程序)可以不加修改地运行于系统结构相同的各档次机器,而且所得结果一致。软件兼容有向上兼容和向下兼容,向上兼容是指低档机器的目标程序(机器语言级)不加修改就可以运行于高档机器上;向下兼容是指高档机器的目标程序不加修改可以运行于低档机器上。一般不使用向下兼容方式。软件兼容还有向前兼容和向后兼容之分。向后兼容是指在某个时期投入市场的该型号机器上编制的软件可不加修改就可以运行于在它之后投入市场的机器上。向前兼容则反之。对系列机而言,必须保证做到软件向后兼容,力争做到软件向上兼容。

计算机系列化有以下优点:

①在使用共同系统软件的基础上解决程序的兼容性问题。

②在统一数据结构和指令系统的基础上,便于组成多机系统和网络。

③使用标准的总线规程,实现接插件和扩展功能卡的兼容,便于实现DEM(由各厂生产功能卡,然后组装成系统)。

④扩大计算机应用领域,提供用户在同系列的多种机型内选用最合适的机器的可能性。

⑤有利于机器的使用、维护和人员培训。

⑥有利于计算机升级换代。

⑦有利于提高劳动生产率,增加产量,降低成本,促进计算机的发展。

系列机为了保证软件的兼容,要求系统结构不准改变,这无疑又成为妨碍计算机系统结构发展的重要因素。实际上,为适应性能不断提高和应用领域不断扩大的需要,应允许系列机中后面推出的各档机的系统结构有所发展和变化。但是,这种改变只应该是为提高机器总的性能所作的必要扩充,而且主要是为改进系统软件的性能来修改系统软件(如编译系统),尽可能不影响高级语言应用软件的兼容,尤其是不允许缩小或删改运行已有软件的那部分指令和结构。例如,在后推出的各档机器上,可以为提高编译效率和运算速度增加浮点运算指令;为满足事务处理增加事务处理指令及其所需功能;为提高操作系统的效率和质量增加操作系统专用指令和硬件等等。因此,可以对系列机的软件向下兼容和向前兼容不作要求,向上兼容在某种情况下也可能做不到(如在低档机器上增加了面向事务处理的指令),但向后兼容是肯定要做到的。

把不同公司厂家生产的具有相同系统结构的计算机称为兼容机。它的思想与系列机的思想是一致的。兼容机还可以对原有的系统结构进行某种扩充,使之具有更强的功能,例如,长城 0520 与 IBM PC 兼容,但有较强的汉字处理功能。

3. 模拟与仿真

系列机解决了在具有相同系统结构的各种机器之间实现软件移植。为了实现软件在不同系统结构的机器之间移植,就必须做到能在一种机器的系统结构上实现另一种机器的系统结构。从计算机系统结构的层次模型来看,就是要在一种机器的系统结构上实现另一种机器的指令系统。一般可采用模拟方法或仿真方法。

如图 1.3 所示,在 A 机的机器语言级上,用虚拟机的概念实现 B 机的指令系统,即 B 机的每一条机器指令由一段 A 机的机器语言程序去解释执行,从而可使 B 机的程序能在 A 机上运行。这种用机器语言程序解释实现程序移

植的方法称为模拟,被模拟的 B 机称为虚拟机,A 机称为宿主机。

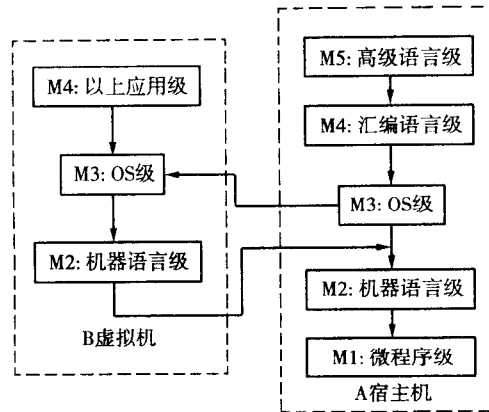


图 1.3 在 A 机器语言级上实现 B 机器指令系统

若 A 机采用微程序控制,则被模拟的 B 机的每条机器指令需要通过二重解释。显然,如果直接用 A 机的微程序去解释 B 机的机器指令就会加快解释过程。这种用微程序直接解释另一种机器的指令系统称为仿真。仿真与模拟的主要区别在于解释所用的语言。仿真是用微程序解释,其解释程序在微程序存储器中。模拟是用机器语言解释,其解释程序在主存储器中。

为了使虚拟机的应用软件能在宿主机上运行,除了模拟虚拟机的机器语言外,还得模拟其存储体系、I/O 系统、控制台的操作,以及形成虚拟机的操作系统,让虚拟机的操作系统受宿主机操作系统的控制,如图 1.3 所示那样,实际上是把虚拟机操作系统作为宿主机的应用程序来看待。所有为模拟所编制的解释程序统称为模拟程序。

模拟程序的编制是非常复杂和费时的。由于虚拟机的每条机器指令不能直接被宿主机的硬件执行,而是由多条宿主机机器指令构成的解释程序来解释,因此,模拟的运行速度显著降低。

用仿真方法可以提高被移植软件的运行速度,但由于微程序机器级结构深深依赖于机器的系统结构,所以当两种机器结构差别较大时,就很难依靠仿真来实现软件移植,特别是当其 I/O 系统结构差别较大时更是如此。因此,在实际应用中,不同系列机之间的软件移植往往通过仿真和模拟两种方法并用来实现。对于使用频繁的易于仿真的机器指令,尽可能采用仿真方法以提高速度,对于使用较少且用微程序仿真难以实现的某些指令及 I/O 系统等操作则宜采用模拟方法。

1.2 计算机系统结构中并行性的发展

计算机系统结构已在冯·诺依曼结构基础上有了很大的改进和发展。一方面合理地进行了软硬功能分配,适当增大硬件实现的功能比例,为操作系统、高级语言和应用软件的实现提供更多更好的支持;另一方面则是通过各种途径来提高计算机系统结构中的并行处理能力,发展全新的计算机系统结构。

1.2.1 冯·诺依曼型计算机的特点与问题

冯·诺依曼型计算机由运算器、控制器、存储器、输入设备和输出设备5部分组成。冯·诺依曼型计算机在结构上有以下特点:

①机器以运算器为中心,I/O设备与存储器之间的数据传送都要经过运算器。各部分的操作及其相互之间的联系都要由控制器集中控制。

②采用存储程序的思想,即机器各部分的操作是在事先存放在存储器中的程序的控制之下,顺序执行一条条指令来完成的。而且将程序的指令和存储器中的数据同等对待,都可以不加区别地送到运算器,因此,由指令组成的程序可以在运行过程中被修改。

③存储器按地址访问,它是一个顺序、线性编址的一维空间,每个存储单元的位数是固定的。

④由指令计数器指明要执行的指令在存储器中的地址,可以根据运算结果改变指令计数器的值,用来改变指令执行顺序。

⑤指令由操作码和操作数地址码两部分组成,操作数的数据类型(如定点数、浮点数、十进制数、双精度数、逻辑数、字符串等等)由操作码指明,操作数不能判定它本身是何种数据类型。

⑥数据以二进制编码,并采用二进制运算。

⑦软件与硬件完全分开,硬件结构采用固定性逻辑,即其功能是不变的,完全依靠编制软件来适应不同的应用需要。

早期的计算机采用冯·诺依曼型结构,但是,随着计算机应用领域的扩大和计算机技术的发展,人们已经清楚地认识到这种结构的问题,并已不断地对这种结构加以改进和发展全新的系统结构。冯·诺依曼型结构的主要问题和改进发展的主要表现有:

①由于机器以运算器为中心,使得低速的输入/输出和高速的运算不得不

互相等待、串行进行。而且由于所有部件的操作由控制器集中控制,使控制器的负担过重,从而严重影响了机器速度和设备利用率的提高。因此,很快将机器的结构改为以主存为中心,让系统的输入/输出与 CPU 的操作并行,多种输入和输出并行,并进一步发展为分布处理和并行处理。

②由于采用存储程序控制的思想,机器各部分的操作是在一条条顺序执行的指令控制下完成的。如果现行指令需要的数据是它前面指令的执行结果,那么顺序执行当然是必须的。但是,实际程序中的许多前后指令的数据互不相关,因此,单纯的顺序执行指令就难以最大限度地发挥系统的并行处理能力,从而会严重影响计算机性能的提高。于是,人们发展了所谓的数据流计算机。在数据流计算机中,只要一条指令所需要的操作数都准备好了,这条指令就马上可被激发执行,完全不需要程序计数器控制。当前指令的执行结果又会激发另一条或另一批指令的执行,指令的执行与指令在程序中出现的次序完全无关。数据流计算机能最大限度地满足程序的并行性。

③虽然指令和数据混存于同一存储器中,由于共用一套存储器外围电路而节省了硬件,也由于对指令和数据不加区别同等对待而简化了存储管理,但是,由于在程序执行过程中,指令可能被当作操作数一样被修改,因此,不利于程序调试和排错,也不利于实现程序的可再入性(Reenterability)和程序的递归调用,不利于指令和数据的并行存取以及在组成上采用重叠、流水方式来提高速度。所以,绝大多数计算机已改为指令在执行过程中不准修改的工作方式,有的机器还将指令和数据分别存放在两个独立编址且可以同时被访问的不同存储器中。

④存储器按地址访问,组成顺序、一维的线性空间。这种存储器组成方式虽然有结构简单、价格便宜、访问速度快等优点,但是,存储器的一维线性空间表示与应用中经常需要的栈、树、图、多维数组等这样一些非线性、多维、离散的数据结构相矛盾。过去都是将这些数据结构经软件变换映射到一维线性空间,不仅使软件复杂,效率降低,而且也不适合对大量数据的并行快速查找。为此作了不少改进。例如,使存储器同时具有按字、字节、位的多种编址方式;采用虚拟存储技术;把单一主存改为多体交叉编址的并行存储器;采用按内容访问的相联存储器实现高速相联查找;采用增设一定数量的通用寄存器来减少访问主存的次数;在 CPU 和主存之间设置高速缓冲存储器(Cache);使计算机具有高级寻址能力的的数据表示等等。

⑤为了能进一步开发利用求解问题和程序隐含的并行性,提高运行的速度和效率,人们将原来 CPU 的顺序执行的组成方式改为先行控制、重叠、流水等组成方式,同时开发指令内、指令间、任务间、作业间不同级别上的并行性,

出现了向量处理机、并行处理机、多处理机、分布处理系统等计算机系统结构,进一步发展以非控制流方式驱动的数据流计算机,并研究发展更为复杂的并行算法。

⑥指令由操作码和地址码两部分组成,操作数并不表示它本身是何种数据类型,而是由操作码指出对何种数据类型的操作数进行操作,因此,每增加一种操作数类型,就要增加一组处理这种数据类型的数据的指令。这样,不仅使指令系统日益庞大复杂,而且由于高级语言中操作符与数据类型无关,操作数的类型是由数据类型说明语句说明的。这种机器指令与高级语言之间存在的语义差距过去只好通过编译程序来弥补,从而加重了编译的负担,增大了辅助开销。为此,人们为计算机系统增设了许多高级数据表示,如自定义数据表示,让每个数据自身带有数据类型标志,使指令具有对多种数据类型操作的通用性,从而简化机器指令系统和编译。

过去曾经认为,指令系统的指令条数越多,机器的功能就会越强。事实上,为了支持操作系统和编译程序等系统软件,指令系统不断地增加了许多专用指令,从而使得机器指令系统越来越大,多至数百条指令,称此类计算机为复杂指令系统计算机(CISC——Complex Instruction Set Computer)。到了70年代中期,已开始感到采用这种日益庞大复杂的指令系统不仅实现起来越来越困难,而且实际上还可能降低整个系统的性能。70年代末,提出和发展了所谓精简指令系统计算机(RISC——Reduced Instruction Set Computer)。

⑦软件与硬件截然分开,硬件结构的完全固定,会使得无法更合理地进行软硬功能分配,因而难以优化系统结构的设计,当求解的问题和应用要求变化时,会使性能价格比明显下降。因此,现在已特别强调软硬结合,进一步发展、采用可以灵活地选择和改变指令系统和结构的动态自适应机器。要求研制出智能计算机系统结构来有效地支持知识和信息处理,对知识进行逻辑推理,特别是能利用经验性知识对不完全确定的事实进行非精确性的推理。

总之,现在的计算机系统结构已在冯·诺依曼型结构基础上不断地改进,发生了很大的变化,不断改进的主要特点是通过各种途径来提高计算机系统结构中的并行处理能力,即使是全新的计算机系统结构也同样离不开发展高度的并行处理能力。

1.2.2 并行性的发展

所谓并行性(Parallelism)是指问题中具有可以同时进行运算或操作的特性。开发并行性的目的是为了能对问题进行并行处理,以提高计算机求解问