

21世纪高等院校计算机教材系列

计算机系统结构

● 田俊峰 等编著



购书可获得增值回报
提供教学用电子教案



机械工业出版社
CHINA MACHINE PRESS



21 世纪高等院校计算机教材系列

计算机系统结构

田俊峰 杜瑞忠 宋鑫 李继民 编著



机械工业出版社

全书共分 7 章, 内容包括: 计算机体系结构的基本概念、指令集系统结构的设计及优化、流水线和指令级并行技术、层次化结构的存储体系、输入/输出系统的概念(包括 I/O 系统设计与性能评测)、多处理体系统、互联网络以及机群系统的基本原理等。本书内容力求深入浅出, 以量化的分析方法全面介绍系统结构的基本原理。全书内容丰富, 重点突出, 编排层次合理。

本书可作为计算机科学与技术专业本科生“计算机体系结构”课程的教材, 也可作为相关专业的研究生教材, 还可作为计算机技术应用和研究人员的参考书。

图书在版编目(CIP)数据

计算机系统结构 / 田俊峰等编著. —北京: 机械工业出版社, 2006.1

(21 世纪高等院校计算机教材系列)

ISBN 7-111-17458-8

I. 计... II. 田... III. 计算机体系结构—高等学校—教材 IV. TP303

中国版本图书馆 CIP 数据核字(2005)第 112053 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 李利健

责任印制: 杨 曦

高等教育出版社印刷厂印刷

2006 年 1 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·14.5 印张·357 千字

0001—5000 册

定价: 21.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68326294

封面无防伪标均为盗版

出版说明

计算机技术是一门迅速发展的现代科学技术，它在经济建设与社会发展中，发挥着非常重要的作用。近年来，我国高等院校十分注重人才的培养，大力提倡素质教育、优化知识结构，提倡大学生必须掌握计算机应用技术。为了满足教育的需求，机械工业出版社组织了这套“21 世纪高等院校计算机教材系列”。

在本套系列教材的组织编写过程中，我社聘请了各高等院校相关课程的主讲老师进行了充分的调研和细致的研讨，并针对非计算机专业的课程特点，根据自身的教学经验，总结出知识点、重点和难点，一并纳入到教材中。

本套系列教材定位准确，注重理论教学和实践教学相结合，逻辑性强，层次分明，叙述准确而精炼，图文并茂，习题丰富，非常适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

参加编写本系列教材的院校包括：清华大学、西安交通大学、北京交通大学、北京邮电大学、北京化工大学、北京科技大学、山东大学、首都经贸大学、河北大学等。

机械工业出版社

前 言

本书是在作者多年从事“计算机体系结构”等课程的教学经验和体会及有关科研成果的基础上,参考国内外优秀教科书编写而成的。它具有结构清晰完整、系统性强、内容丰富而又重点突出、行文由浅入深和可读性强等特点。本书所采用的量化分析方法,能够使读者深入、扎实地掌握和理解系统结构的基本概念和原理,同时培养一种科学而有效的学习和科研的方法。本书所介绍的设计方法和分析技术可以在相当长的一个时期内指导设计者根据当时的技术现状,设计出高性能的系统。

“计算机体系结构”是计算机领域的一门重要学科。它强调从总体结构、系统分析这一角度来研究计算机系统。通过学习本书,可培养读者系统地、自上而下地分析和解决问题的能力。能把读者在“计算机组成原理”、“操作系统”、“编译原理”、“汇编语言程序设计”等课程中所学的软、硬件知识有机地结构起来,从而建立起计算机系统的完整概念。

本书除了着重论述体系结构的基本概念、基本原理、基本结构和基本分析方法以外,还强调采用量化的分析方法。这种方法使我们能更具体、实际地分析和设计计算机体系结构。书中还选用了许多典型例题说明如何进行量化分析。

本书共 7 章,内容包括:计算机体系结构的基本概念、指令系统、流水线技术、存储系统、输入/输出系统、多处理机、机群系统等。

本书由田俊峰、杜瑞忠、宋鑫、李继民编著。

本书可作为计算机科学与技术专业本科生“计算机体系结构”课程的教材,也可作为相关专业的研究生教材,还可作为计算机技术应用和研究人员的参考书。

为了配合教学,本书为读者提供了电子教案,读者可在机械工业出版社网站 <http://www.cmpbook.com> 上下载。

限于作者水平,书中错误和不当之处在所难免,敬请读者批评指正。

作 者

目 录

出版说明

前言

第 1 章 计算机体系结构的基本概念	1
1.1 引论	1
1.2 计算机体系结构的概念	1
1.2.1 计算机系统层次概念	1
1.2.2 计算机体系结构定义	3
1.2.3 计算机组成与实现技术	4
1.3 计算机体系结构的发展	5
1.3.1 存储程序计算机体系结构及其发展	6
1.3.2 计算机的分代和分型	8
1.3.3 应用需求的发展	10
1.3.4 计算机实现技术的发展	11
1.4 计算机体系结构中并行性的发展	11
1.4.1 并行性概念	12
1.4.2 提高并行性的技术途径	12
1.5 定量分析技术基础	14
1.5.1 计算机性能的评测	14
1.5.2 系统结构的评测标准	15
1.5.3 性能设计和评测的基本原则	17
1.5.4 CPU 的性能	19
1.6 小结	22
1.7 习题	22
第 2 章 指令系统	24
2.1 指令集结构的分类	24
2.1.1 指令集结构分类	24
2.1.2 通用寄存器型指令集结构的分类	26
2.2 寻址技术	27
2.3 指令集结构的功能设计	30
2.3.1 CISC 计算机指令集结构的功能设计	31
2.3.2 RISC 计算机指令集结构的功能设计	32
2.3.3 控制指令	34
2.4 操作数的类型、表示和大小	35
2.5 指令集格式的设计	37
2.5.1 寻址方式的表示方法	37

2.5.2 指令集格式的选择	37
2.6 指令格式的优化	38
2.6.1 操作码优化表示	41
2.6.2 地址码优化表示	44
2.6.3 定长指令字操作码同地址码配合	44
2.7 指令系统分析	46
2.7.1 IBM 370 指令系统	46
2.7.2 增设辅助操作码	50
2.7.3 用一条指令代替一串指令	51
2.7.4 对高级语言优化的改进	52
2.8 小结	54
2.9 习题	55
第3章 流水线技术	56
3.1 先行控制技术	56
3.1.1 指令的执行方式	56
3.1.2 先行控制方式的原理	57
3.2 流水线概述	58
3.2.1 流水线的基本概念	58
3.2.2 流水线的分类	60
3.3 流水线中的相关	61
3.3.1 流水线性能分析	61
3.3.2 流水线的结构相关	67
3.3.3 流水线的数据相关	67
3.3.4 流水线的控制相关	70
3.4 非线性流水线	75
3.4.1 非线性流水线的表示	75
3.4.2 非线性流水线的调度技术	77
3.5 超标量和超流水线处理机	80
3.5.1 超标量处理机	80
3.5.2 超流水线处理机	83
3.6 向量处理机	86
3.6.1 向量处理方式和向量处理机	86
3.6.2 向量处理机实例分析	89
3.7 小结	89
3.8 习题	90
第4章 存储系统	91
4.1 存储系统原理	91
4.1.1 存储系统定义	91
4.1.2 存储系统的性能指标	93

4.1.3 频带平衡	94
4.1.4 并行存储器	95
4.1.5 交叉访问存储器	96
4.2 高速缓冲存储器 (Cache)	99
4.2.1 基本工作原理	100
4.2.2 地址映像与变换方法	101
4.2.3 替换算法及其实现	110
4.2.4 Cache 的性能分析	113
4.3 虚拟存储器	117
4.3.1 虚拟存储器工作原理	117
4.3.2 地址的映像与变换	118
4.3.3 加快内部地址变换的方法	123
4.3.4 页面替换算法及其实现	128
4.3.5 提高主存命中率的方法	132
4.4 习题	134
第 5 章 输入/输出系统	137
5.1 概述	137
5.2 存储设备类型	137
5.2.1 磁盘	137
5.2.2 磁盘的未来	139
5.2.3 使用 DRAM 作磁盘	140
5.2.4 光盘	141
5.2.5 磁带	141
5.2.6 自动磁带库	141
5.3 中断系统	142
5.3.1 中断系统的软硬件功能分配	142
5.3.2 中断屏蔽	144
5.4 总线——连接 I/O 设备到 CPU 和存储器	148
5.4.1 总线设计决策	149
5.4.2 总线标准	151
5.4.3 存储设备与 CPU 的接口	152
5.4.4 CPU 授权 I/O 操作	153
5.5 通道处理机	154
5.5.1 通道的作用和功能	154
5.5.2 通道的工作过程	155
5.5.3 通道种类	156
5.5.4 通道中的数据传送过程	159
5.5.5 通道的流量分析	160
5.6 I/O 性能评测	161

5.7 可靠性、可用性和 RAID	163
5.7.1 镜像 (RAID1)	164
5.7.2 位交叉奇偶校验 (RAID 3)	164
5.7.3 块交叉分布奇偶校验 (RAID 5)	165
5.8 I/O 系统设计	165
5.9 习题	166
第 6 章 多处理机	168
6.1 简介	168
6.2 并行系统结构的分类	168
6.3 多处理机的性能模型	170
6.3.1 基本模型	171
6.3.2 N 台处理机系统的基本模型	172
6.3.3 多处理机模型	173
6.4 多处理机的 Cache 一致性	174
6.4.1 Cache 一致性问题的由来	174
6.4.2 监听协议	176
6.4.3 基于目录的协议	179
6.5 互连网络	182
6.5.1 互连网络的性能参数	182
6.5.2 静态连接网络	185
6.5.3 动态连接网络	189
6.6 多处理机实例	192
6.6.1 MPP	193
6.6.2 对称多处理机	195
6.7 习题	200
第 7 章 机群系统	201
7.1 机群系统概述	201
7.2 机群系统的组成和特点	201
7.2.1 机群系统的组成	201
7.2.2 机群系统的特点	202
7.2.3 微机机群技术	203
7.2.4 微机机群的可靠性	205
7.2.5 微机机群的优势	206
7.2.6 微机机群的局限	207
7.2.7 微机机群的发展趋势	208
7.3 机群系统的通信技术	209
7.4 并行程序设计环境 (MPI)	213
7.4.1 MPI 的基本概念	214
7.4.2 MPI 的特点	215

7.4.3 目前主要的 MPI 实现	216
7.5 机群系统的负载均衡技术	216
7.5.1 负载均衡的基本问题	216
7.5.2 负载均衡系统的主要技术	217
7.5.3 负载均衡系统设计的考虑因素	218
7.5.4 机群系统负载均衡技术的研究状况	220
7.6 习题	221
参考文献	222

第1章 计算机体系结构的基本概念

1.1 引论

自1946年第一台电子数字计算机问世以来,计算机已经经历了五次更新换代。换代的标志主要有两个:其一是构成计算机的器件。几十年来,器件发生了很大的变化,经电子管、晶体管发展到集成电路,而集成电路又由小规模、中规模到大规模及超大规模阶段。器件的更新,主要表现为速度、功能、集成度、可靠性的不断提高和成本的不断降低,是计算机发展的物质基础。因此,器件的更新是计算机换代的最突出标志;其二是系统结构的特点。系统结构不断改进,许多重要概念不断提出并且得到实现。例如,变址寄存器概念、通用寄存器概念、浮点数据表示概念、程序中断概念、输入/输出通道概念等。很明显,如果用大规模集成电路实现早期的计算机系统结构,人们并不会认为它是第四代计算机。因为第四代计算机在系统结构上较早期的计算机已经有了很大的改进和发展。因此,系统结构方面的特点同样是计算机换代的重要标志。

回顾计算机的发展历史,可以看出,计算机系统性能的不断提高主要靠器件的变革和系统结构的改进。恩斯洛(P.H.Enslow)曾比较了1965~1975年这十年间,器件延迟时间和计算机指令时间的关系。结果表明,这十年间器件延迟时间降低至原来的十分之一,但计算机的指令时间却是原来的百分之一。这种情况在计算机近年来的发展中变得更加明显。因此,如何最合理地利用新器件、最大限度地发挥其潜力、设计并构成综合性能指标最佳的计算机系统,单纯依靠器件变革是不能实现的,还要靠计算机系统结构上的改进。

本书专门研究计算机系统结构,重点在计算机体系结构的设计与分析。本章将解决:什么是计算机体系结构,计算机体系结构和技术有什么关系,计算机体系结构中的并行性等问题。

1.2 计算机体系结构的概念

设计一种新型计算机系统必须面临的问题有很多,其中最主要的是新型计算机的主要特点和性能。它们具体包括:指令集设计、功能组织、逻辑设计、实现技术等。实现技术又包括集成电路设计、制造和封装技术、系统制造、供电、冷却技术等。另外,我们往往要求在限定的造价范围内,使新型计算机具有最高的性能。如何采用先进的计算机体系结构和生产技术制造出具有高性能价格比的计算机系统,是所有通用计算机设计的共同目标。

1.2.1 计算机系统中的层次概念

从计算机系统的研究分析、开发生产与应用角度来看,计算机系统可以按其功能特性的不同划分为多个层次,每个层次的功能主要表现在其语言的功能上。

随着计算机系统的发展，人们已经设计出一系列语言。从面向机器的语言（机器指令系统、汇编语言）到面向过程的各种高级程序设计语言（C/C++语言、Pascal），到面向问题的语言或者说叫应用语言（面向数据库查询的 SQL 语言）。计算机语言就是这样由低级向高级发展，低一级语言是高级语言产生的基础，高级语言由低一级语言发展而来，它比低一级语言功能要强，便于应用。根据这一系列的多级计算机语言，我们可以把计算机系统划分为如图 1-1 所示的多层结构。

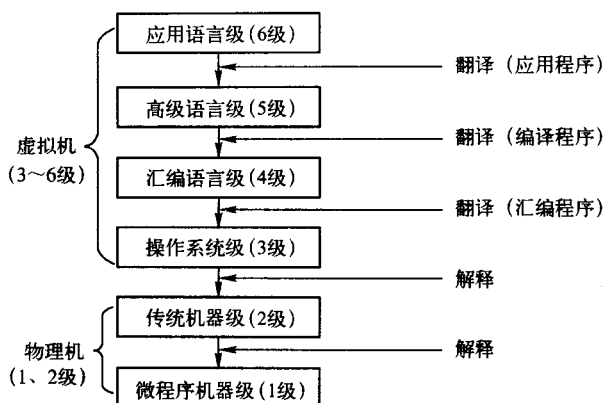


图 1-1 计算机系统的层次结构

其中，第 1 级和第 2 级的功能是由硬件或固件实现的，被称为物理机。第 3~6 级的功能是由软件实现的，被称为虚拟机。

第 1 级是微程序机器级，它的机器语言是微指令集，各种指令由微程序控制执行。微程序控制器是由固件实现的。

第 2 级是传统机器级，它的机器语言是该机的指令系统。用这级机器语言编写的程序是通过微程序进行解释的。这个解释程序运行在第 1 级上。由微程序解释指令系统又称作仿真。在第 1 级上可以有一个或多个能在它上面运行的解释程序，每一个解释程序都定义了一种指令系统。因此，可以通过仿真在一台机器上实现多种指令系统。

应该注意的是：在计算机系统中，可以没有微程序机器级，而用硬件直接实现机器的指令集。用硬件直接实现机器的指令集可以大大提高指令的执行速度。

第 3 级是操作系统级。从操作系统的基本功能看，一方面，它要管理传统机器的软、硬件资源；另一方面，它又是传统机器的引申。这级语言中的某些指令就是第 2 级传统机器级的指令，如：算术运算、逻辑运算等类型的指令。另外，它又提供了传统机器所没有的，但为汇编语言和高级语言的使用和实现所需的某些操作指令和数据结构，如文件结构和文件管理的基本操作（打开文件、读/写文件、关闭文件的指令）。

程序员用这一级语言编写的程序，一部分由第 2 级进行解释。运行在第 2 级上的解释程序习惯上叫做操作系统。而那些与第 2 级指令相同的第 3 级指令是直接由微程序实现的。

第 4 级是汇编语言级。这级的机器语言是汇编语言。用汇编语言编写的程序，首先翻译成第 2 级或第 3 级语言，再由相应的机器进行解释。完成翻译的程序叫做汇编程序。

第 4 级改变了前面几级所使用的解释的实现方法，而采用翻译的方法来实现。解释和翻

译是语言实现的两种基本技术，它们都是以执行一串 N 级指令来实现 $N+1$ 级指令。但二者仍存在着差别，翻译技术是先把 $N+1$ 级程序全部变成 N 级程序后，再去执行新产生的 N 级程序。在执行过程中， $N+1$ 级程序不再被访问。而解释技术是每当 $N+1$ 级指令被译码后，就直接去执行一串等效的 N 级指令，然后再去取下一串 $N+1$ 级指令，依此重复执行。在这个过程中不产生翻译出来的程序，因此，解释的过程是一边变换一边执行的过程。在实现新的虚拟机器时，这两种技术都被广泛使用。一般来说，解释执行比翻译花的时间多，但占用存储空间较少。

第 5 级是高级语言级。这级的机器语言就是各种高级语言。用这些语言编写的程序一般由编译程序翻译到第 3 级或第 4 级上 (C/C++)。个别高级语言也用解释的方法实现 (BASIC 语言系统)。

第 6 级是应用语言级。这一级是为了使计算机满足某种用途专门设计的，因此，这一级的语言就是面向问题的各种应用语言。可以设计专门用于人工智能、行政、教育等方面的虚拟机。这些虚拟机也是当前计算机应用领域的重要的研究课题。应用语言编写的程序一般由应用程序包翻译到第 5 级上。

由上所述，我们可以把计算机理解为如图 1-2 所示的一台虚拟机。

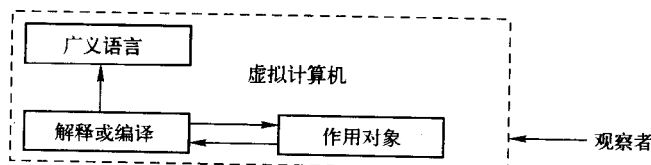


图 1-2 机器的作用和组成

把复杂的计算机系统按功能划分为多级层次结构具有以下优点：

- 有助于我们理解计算机系统的工作，明确软、硬件及固件在计算机系统中的地位和作用。
- 有助于我们理解各种语言的实质和实现。
- 对于探索新的虚拟机的实现方法、设计新的计算机系统也是有好处的。
- 有助于我们理解计算机体系结构的定义，从而进行更合理的计算机系统结构的设计。

1.2.2 计算机体系结构定义

“计算机体系结构”这个名词来源于英文 Computer Architecture，也有译成“计算机系统结构”的。Architecture 这个词原来用于建筑领域，其意义是“建筑学”、“建筑物的设计或式样”，它指一个系统的外貌。20 世纪 60 年代，这个名词被引入计算机领域，被译为“构造、结构”等。“计算机体系结构”一词已经得到普遍应用，它研究的内容不但涉及计算机硬件，也涉及计算机软件，已成为一门学科。但对“计算机体系结构”一词的含义仍有多种说法，并无统一的定义。

经典的“计算机体系结构”定义是 1964 年 C. M. Amdahl 在介绍 IBM 360 系统时提出的：计算机体系结构是程序员所看到的计算机的属性，即概念性结构与功能特性，这些属性既不同于数据流和控制的组织，也不同于逻辑设计和物理实现。

按照计算机系统的多层次结构,不同层次的计算机程序员所看到的计算机具有不同的属性。例如,传统机器级程序员所看到的计算机的主要属性是该机指令系统的功能特性。而高级语言级程序员看到的计算机的主要属性是该机所配置的高级语言所具有的功能特性。但是,不同的计算机系统,它们的属性的不同只是表现在高级语言级以下,从高级语言级程序员的角度看,各种计算机系统具有相同的属性,它们没什么不同。在层次结构中,各层有各自不同的体系结构,Amdahl所说的计算机体系结构是指传统机器级的体系结构,即一般所说的机器语言程序员所看到的传统机器级所具有的属性。

这些属性是计算机设计者(或者编译程序生成系统)为使其所设计(或生成)的程序能在机器上正确运行,所需看到和遵循的计算机属性。这些属性不仅和处理机有关,而应是工作在传统机器上的程序员所看到的机器的所有部分,包括处理机、存储系统、I/O 联结方式和中断机构等。但不包括机器内部的数据流和控制流,以及逻辑设计和器件设计等。

目前,对于通用寄存器型机器,具有以下属性:

- 数据表示 硬件能直接辨认和处理哪些数据。
- 寻址规则 包括最小寻址单元、寻址方式,以及寻址方式的表示。
- 寄存器定义 包括操作数寄存器、变址寄存器和控制寄存器等的定义、数量和使用方式。
- 指令系统 包括机器指令的操作类型和格式、指令间的排序和控制机构等。
- 中断机构 中断的类型和中断响应硬件的功能等。
- 机器工作状态(如管态和目态等)的定义和切换。
- 存储系统 主存容量、程序员可用的最大存储容量等。
- 信息保护 包括信息保护方式和硬件对信息保护的支持。
- I/O 结构 包括 I/O 联结方式、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,以及 I/O 操作的结束与出错指示等。

在计算机技术中,一种本来是存在的事物或属性,但从某种角度看似乎不存在,称为透明性现象。通常,在一个计算机系统中,低层机器的属性对高层机器的程序员往往是透明的,如传统机器级的概念性结构和功能特性,对高级语言程序员来说是透明的。由此看出,在层次结构的各个级上都有它的体系结构。Amdahl 提出的体系结构是指传统机器级的体系结构,即一般所说的机器语言程序员所看到的传统机器级所具有的属性。

这里讨论的是经典的计算机体系结构的概念。计算机体系结构作为一门学科,主要研究软、硬件功能分配,以及对软、硬件界面的确定,即哪些功能由软件完成,哪些功能由硬件完成。随着计算机技术的发展,计算机体系结构所包含的内容是不断变化和发展的。目前经常使用的是广义的计算机体系结构概念,它既包括经典的计算机体系结构的概念范畴,还包括对计算机组成和计算机实现技术的研究。

1.2.3 计算机组成与实现技术

计算机组成(computer organization)也常称为计算机组织。计算机系统机构确定了分配给硬件子系统的功能及其概念之后,计算机组成的任务是研究各组成部分的内部构造和相互联系,以实现机器指令级的各种功能和特性。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,是计算机组织合理的重要标志,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,人们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、多模块交叉工作、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,人们研究出通道、外围处理机等方式。为了提高处理机速度,人们研究出先行控制、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。

计算机组成是计算机系统结构的逻辑实现,包括机器内部的数据流和控制流的组成以及逻辑设计等。计算机组成的设计是按照所希望的性价比,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。一般的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、缓冲器和排队的确定、控制机构的设计、可靠性技术的确定等。对传统的机器程序员来说,计算机组成的设计内容一般是透明的。

计算机实现是指计算机组成的物理实现。它包括处理机、主存等部件的物理结构,器件的集成度和速度,信号传输,器件、模块、插件、底板的划分与连接,专用器件的设计,电源、冷却、装配等技术以及有关的制造工艺和技术等。

计算机系统结构、计算机组成和计算机实现是三个不同的概念。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。

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。有些计算机系统是作为系统结构的内容,其他计算机系统可能是作为组成和实现内容。例如,高速缓冲存储器一般是作为组成提出来的,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,有的机器为了提高其使用效率,设置了高速缓冲存储器的管理指令,使程序员能参与高速缓冲存储器的管理。这样,高速缓冲存储器又成为系统结构的一部分,对程序员来说是不透明的。

Amdahl 等人的计算机系统结构定义的主要内容是指令系统及其执行模型。根据这个定义,一个系列机中不同档次的机器有相同的系统结构。Amdahl 等人定义系统结构时认为,只要指令系统兼容就能保证程序正确运行。由于程序的执行要依赖于程序库、操作系统和其他 Amdahl 等人的系统结构定义中没有涉及的因素,这要求操作系统接口等其他层次标准化。同时,由于 VLSI 的迅速发展及其成本急剧下降,有些系列机推出含有新指令的机器,例如,24 位地址的 IBM 360 和 IBM 370 系统发展为 31 位地址的 370 x A 系统,16 位地址的 PDP11 发展为 32 位地址的 VAX 系列。随着新器件的出现,当今计算机设计者面临的问题与十年前面临的问题大不相同,所以我们应当把计算机系统结构定义得更宽一些,除了 Amdahl 等人定义的内容外,还应包括功能模块的设计。也就是说,计算机系统结构、计算机组成、计算机实现之间的界限越来越模糊了。

1.3 计算机体系结构的发展

计算机体系结构研究计算机系统中软、硬件的界面,即研究哪些功能由软件完成,哪些功能由硬件完成。实际上,软件和硬件在逻辑功能上是等效的。就是说,由软件实现的功能

在原理上可以由硬/固件实现。同样，由硬件实现的功能原理上也可以通过软件模拟来实现。但是，软件和硬件在性能上是不等效的。因此，对于计算机系统软/硬件功能分配应保证在满足应用的前提下，充分利用硬件和器件技术的发展，使系统达到较高的性能价格比。

1.3.1 存储程序计算机体系结构及其发展

最早的存储程序计算机是由著名数学家冯·诺依曼等人在 1946 年总结并明确提出来的，因此，经常称之为冯·诺依曼结构计算机。它由运算器、控制器、存储器和输入/输出设备组成，如图 1-3 所示。

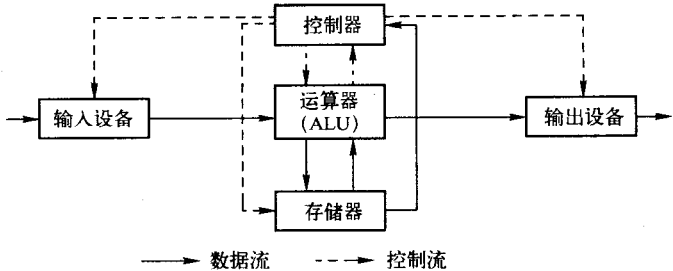


图 1-3 存储程序计算机的结构

现代的计算机体系结构与冯·诺依曼等人当时提出的计算机体系结构相比，虽已发生了重大变化，但就其结构原理来说，占有主流地位的仍是以存储程序原理为基础的冯·诺依曼型计算机。

存储程序计算机在体系结构上的主要特点如下：

- 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途经运算器。运算器、存储器、I/O 设备的操作以及它们之间的联系都由控制器集中控制。
- 采用存储程序原理。程序（指令）和数据放在同一个存储器中，并且没有对两者加以区分。指令和数据一样可以送到运算器进行运算，即由指令组成的程序本身是可以修改的。
- 存储器提供可按地址访问的一级地址空间，每个地址是惟一定义的。
- 指令在存储器中按其执行的顺序存储，指令的执行是顺序的，程序分支由转移指令控制。
- 指令由操作码和地址码组成，操作码指明本指令的操作类型，地址码指明操作数和操作结果的地址，操作数的类型由操作码决定，操作数本身不能判定其数据类型。
- 数据以二进制码表示，采用二进制运算。它主要面向数值计算和数据处理。

存储程序计算机体系结构的这些特点奠定了现代计算机发展的基础，但几十年来，人们对这种计算机体系结构进行了许多改进。

1. 分布的 I/O 处理能力

存储程序计算机以运算器为中心，所有部件的操作由控制器集中控制，这一特点带来了慢速 I/O 操作占用快速运算器的问题。此时的 I/O 操作和运算器操作只能串行，互相等待，大大影响了运算器效率的发挥。尤其是计算机的应用进入商业领域之后，数据处理要求频繁地进行 I/O 操作，上述缺点显得更加突出。

为了克服这一缺点，人们提出了各种输入/输出方式，如图 1-4 所示，从而使计算机概念性结构发生了相应的变化。

在程序等待方式之后，很快出现了程序中中断的概念，并被应用于输入/输出操作。这时 CPU 执行到一条输入/输出指令后，可以不必等待外围设备回答的状态信息而继续执行后续指令，直到外围设备准备好发送/接收数据，向 CPU 发出中断请求，CPU 响应中断请求后才进行输入/输出。这时，计算机结构仍以运算器为中心，但增加了中断接口部件。程序中中断的引入可以使 CPU 与外围设备在一定程度上并行工作，提高了计算机的效率，并且可以实现多种设备的同时工作。中断技术已经成为现代计算机操作系统的技术基础。

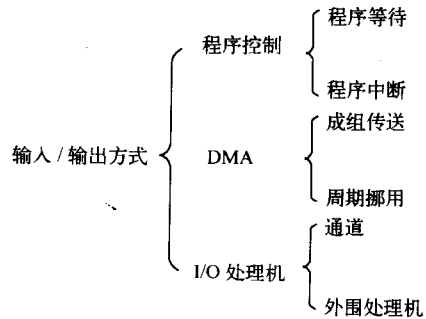


图 1-4 各种输入/输出方式

随着外围设备种类和数量的增多，中断变得过于频繁，从而浪费了大量的 CPU 时间。于是，对于成块（组）地进行传送的输入/输出信息，出现了所谓的 DMA（直接存储器访问）方式。为了实现这种方式，需要在主存和设备之间增加 DMA 控制器（数据通道），从而形成了以主存为中心的结构。CPU 向 DMA 控制器寄存器设置好初始参数后，仍可继续执行其后续指令。当外设准备好发送或接收数据时，便对 CPU 发出 DMA 请求，使输入/输出设备经数据通道直接与主存成组地交换信息。数据通道控制完成这一块数据传送后，向 CPU 发出结束信号，CPU 就可以进行接收或发送数据后的处理工作。

采用 DMA 方式，每传送完一组数据，就要中断 CPU 一次。如果该部件能自己控制完成输入/输出的大部分工作，则 CPU 可进一步摆脱用于管理、控制 I/O 系统的沉重负担。这就出现了 I/O 处理机方式。I/O 处理机几乎把控制输入/输出操作和信息传送的所有功能都从 CPU 那里接管过来。I/O 处理机方式有通道方式和外围处理机（I/O 子系统）方式两种。

2. 保护的存储器空间

虽然传统存储程序计算机的存储程序原理现在仍为大多数计算机所采用，但是是否把指令和数据放在同一存储器中，不同计算机却有不同考虑。将指令和数据存放在同一个存储器中，会带来如下好处：指令在执行过程中可以被修改，因而可以编写出灵活的可动态修改的程序；对于存取指令和数据，仅需一套读写和寻址电路，硬件简单；不必先区分指令和数据，存储管理软件容易实现；程序和数据可以分配于任何可用空间，从而能更有效地利用存储空间等。也有不足之处是：自我修改程序是难以编制、调试和使用的；一旦程序出错，进行程序诊断也不容易；程序的修改不利于实现程序的可再入性（reenterability）和程序的递归调用。在开发指令级并行时还可以看到：程序可修改也不利于重叠和流水方式的操作。因此，现在绝大多数计算机都规定，在执行过程中不准修改程序。这是通过存储管理硬件的支持，由操作系统软件实现的，目前几乎所有支持多任务的计算机操作系统都实现了这种管理。

3. 存储器组织结构的发展

按地址访问的存储器具有结构简单、价格便宜、存取速度快等优点。但是在数据处理时，往往要求查找具有某种内容特点的信息。在随机存储器中，虽然可以通过软件按一定的算法（顺序查找、对分查找或采用 hash 技术等）完成查找操作，但会由于访问存储器的次数较多而影响计算机系统的性能。按内容访问的相联存储器，则把查找、比较的操作交内存存储器