

计算机等级考试教程

新大纲

计算机体系结构

刘惠珍 方 娟 等编著

四级

- 资深专家最新修订版本
- 一至四级整体解决方案
- 兼顾教学、培训及考前复习
- 习题陪练加强效果

全国高等学校计算机教育研究会 组编
课程与教材建设委员会

李大友 主编



机械工业出版社
China Machine Press

计 算 机 等 级 考 试 教 程

(四级)

计算机体系结构

全国高等学校计算机教育研究会
课 程 与 教 材 建 设 委 员 会 组编

李大友 主编

刘惠珍 方 娟 等编著

机 械 工 业 出 版 社

本教材系统地讲述了计算机系统结构的基本概念、基本原理和基本设计分析方法及有关技术的发展,有些技术结合 RISC 和奔腾 II/III 作了一些说明,每章后面均附有部分习题供参考。

全书共分 8 章。主要内容包括:计算机系统结构概念、设计原则、性能评价和发展;存储系统的层次结构和存储器性能分析;输入输出系统中重点介绍通道和 I/O 处理机;数据表示和指令系统的设计分析方法及 RISC 计算机结构;流水处理技术介绍流水的概念、性能及调度方法,超级计算机及向量流水处理机;并行处理技术,包括 SIMD 并行机及 MIMD 多处理机系统;还介绍了互连网基本概念,互连网络性能以及非·冯诺依曼计算机。

本书是按最新全国计算机等级考试(四级)的计算机系统结构大纲编写的,可作为等级考试教材使用,也可作为高等学校计算机系统结构课程教材使用。

图书在版编目(CIP)数据

计算机体系结构/李大友主编.—北京:机械工业出版社,2001,1
计算机等级考试教程.四级
ISBN 7-111-08722-4

I.计… II.李… III.计算机体系结构—水平考试—教材 IV.TP303

中国版本图书馆 CIP 数据核字(2001)第 02213 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:何文军 版式设计:张世琴 责任校对:韩 晶

封面设计:姚 毅 责任印制:路 琳

成都新华印刷厂印刷·新华书店北京发行所发行

2001 年 5 月第 1 版·第 1 次印刷

787mm×1092mm¹/16·19.75 印张·487 千字

0 001—4 000 册

定价:32.00 元

凡购本书,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) 68993821、68326677-2527

前 言

随着微电子技术和超大规模集成电路技术的发展,系统结构新技术也不断涌现。早期巨型机、大型机所特有的提高系统性能的技术已融入到小型机和微型机产品中。因此,系统结构已经成为计算机专业人员必须掌握的知识。

本书介绍了计算机系统结构的基本概念及为提高计算机性能/价格比所涉及的主要技术和计算机系统的基本设计分析方法,为反映计算机系统结构的发展,对于精简指令系统计算机(RISC)及并行处理技术和多处理机系统也做了深入论述,对于新型计算机系统也做了简要说明。

本书内容以系统结构并行技术发展为主线。首先,介绍计算机系统结构的基本概念和主要内容,接着介绍高性能单处理机中涉及的几个子系统的设计和分析方法(存储体系、I/O系统、指令系统及RISC),然后介绍SIMD的流水技术及并行处理机,再进一步发展到MIMD多处理机系统,最后介绍非冯·诺依曼计算机结构。全书共分8章,具体安排如下:

第1章绪论,主要介绍了计算机系统的概念、层次结构、系统结构设计及发展和计算机性能评价;第2章存储系统,主要介绍存储系统的概念及性能分析,高速缓冲存储器功能设计和性能分析,虚拟存储器及存储保护;第3章输入输出系统,介绍了输入输出系统特点发展,并重点介绍通道处理机和输入输出处理机;第4章数据表示和指令系统,主要介绍计算机的数据表示与数据结构的关系,寻址方式的分析,指令系统的设计和优化,并重点介绍了RISC计算机指令系统;第5章流水处理机技术,主要介绍流水的概念及相关处理,流水调度;向量的流水处理与向量处理机,并举例说明Pentium II/III的流水结构;第6章并行处理机技术和多处理机系统。主要介绍并行处理技术及发展,并行软件;SIMD并行处理机的结构、特点及相应算法。MIMD多处理机结构特点及多处理机系统举例;第7章互连网络,主要介绍了互连网络的概念、基本特性、结构,多处理机的互连网络的举例;第8章非冯·诺依曼机结构,主要通过对Von Neumann型计算机的分析,重点介绍基于数据驱动的数据流计算机的原理和性能、并简单介绍了归约机及人工智能计算机的基本原理和结构。每章后面均附有练习题。

系统结构的先修课为数字逻辑、计算机原理、数据结构。它是连接硬件和软件的课程,本身自成体系,可作为全国计算机等级考试教材使用,也可作为高等学校计算机系统结构教材使用。

本书由李大友教授主编,其中第1、2、3章由葛本修教授编写,第4、6、8章由刘惠珍教授编写,第5、7章由方娟老师编写,最后由刘惠珍、方娟老师统一加工修改。本书部分章节文字资料整理由陈帆、赵琉涛等同志协助完成,在此一并表示感谢。

由于计算机系统结构发展迅速,许多新概念、新技术不断涌现,再加作者水平所限,书中定有许多不当之处,敬请广大读者批评指正。

编 者

目 录

前 言

第1章 概论	1
1.1 现代计算机系统	1
1.1.1 现代计算机的构成	1
1.1.2 计算机系统的层次结构	2
1.2 计算机系统结构	3
1.2.1 计算机系统结构的基本概念	3
1.2.2 计算机系统结构、组成与实现	4
1.2.3 计算机系统结构的分类	6
1.3 计算机系统设计	8
1.3.1 计算机系统的设计原则	8
1.3.2 计算机系统结构设计	9
1.4 计算机的性能评价	11
1.4.1 CPU 性能	12
1.4.2 MIPS 和 MFLOPS	13
1.4.3 系统性能测试	14
1.5 计算机系统结构的发展	16
1.5.1 冯·诺依曼机型及其演变	16
1.5.2 系统结构并行技术的发展	18
1.5.3 影响计算机系统结构的因素	19
习题	21
第2章 存储系统	25
2.1 存储系统概述	25
2.1.1 存储系统的形成	25
2.1.2 存储系统的层次结构	26
2.1.3 存储系统的局部性、包含性、 一致性	28
2.2 并行存储器	28
2.2.1 多体并行访问存储器	29
2.2.2 多体并行交叉访问存储器	30
2.3 高速缓冲存储器(Cache)	35
2.3.1 高速缓冲存储器的功能、结构与 工作原理	35
2.3.2 地址映像与转换	37

2.3.3 替换策略	42
2.3.4 Cache 的写操作	45
2.3.5 Cache 性能分析	47
2.3.6 Cache 的实用举例	49
2.4 虚拟存储器	51
2.4.1 概述	51
2.4.2 虚拟存储器的管理方式	52
2.4.3 虚拟存储器的工作过程	58
2.4.4 页面替换策略	60
2.4.5 虚拟存储器的性能分析	62
2.5 存储保护	63
2.5.1 存储保护的方式	63
2.5.2 虚拟存储器与存储保护举例	65
习题	67
第3章 输入输出系统	73
3.1 概述	73
3.1.1 什么是输入输出系统	73
3.1.2 输入输出系统的特点	74
3.2 输入输出系统工作原理	75
3.2.1 I/O 系统的结构	75
3.2.2 I/O 系统的逻辑组成及工作原理	75
3.2.3 总线技术	77
3.3 通道处理机	83
3.3.1 通道的功能	83
3.3.2 通道的逻辑组成与工作过程	84
3.3.3 通道指令与通道程序	86
3.3.4 输入输出指令	89
3.3.5 输入输出中断	90
3.3.6 通道的种类及流量分析	90
3.4 输入输出处理机	95
3.4.1 输入输出处理机的功能	95
3.4.2 输入输出处理机系统举例	95
习题	96

第4章 数据表示和指令系统	99	5.3.2 非线性流水线调度	173
4.1 概述	99	5.4 流水处理中指令并行性 进一步开发	175
4.1.1 计算机指令系统的设计 过程和要求	99	5.4.1 超级标量计算机	175
4.1.2 指令系统发展改进的两种 途径 CISC, RISC	100	5.4.2 超长指令字计算机	177
4.2 数据类型和数据表示	100	5.4.3 超级流水线计算机	178
4.2.1 数据类型	100	5.4.4 超标量超流水线计算机	178
4.2.2 基本数据表示	101	5.4.5 奔腾系列处理器流水处理 举例	180
4.2.3 二进制的定点、浮点数据表示	102	5.5 向量的流水处理	183
4.2.4 自定义数据表示和向量 数据表示	103	5.5.1 向量处理方式	183
4.3 指令系统设计原理	109	5.5.2 向量处理机的结构	184
4.3.1 指令格式和指令系统的分类	109	5.5.3 改进向量处理机性能的方法	187
4.3.2 寻址技术	112	5.5.4 向量处理机的性能	190
4.3.3 指令系统功能设计	113	习题	193
4.3.4 指令格式的优化	114	第6章 并行处理技术和多处理机	197
4.3.5 指令系统的执行和优化	121	6.1 概述	197
4.4 RISC 计算机	128	6.1.1 并行性概念	197
4.4.1 RISC 计算机设计原理	128	6.1.2 并行的等级和分类	197
4.4.2 RISC 计算机体系结构和 指令系统	131	6.1.3 并行处理的应用	200
4.4.3 RISC 的主要技术	133	6.2 并行处理技术及发展	200
4.4.4 RISC 计算机举例	136	6.2.1 并行处理技术	200
4.4.5 RISC 计算机的发展	141	6.2.2 并行处理中需研究的课题	203
习题	142	6.3 SIMD 并行处理机	204
第5章 流水线处理技术	144	6.3.1 SIMD 并行处理机的基本构成	204
5.1 概述	144	6.3.2 SIMD 并行处理机算法	211
5.1.1 处理器结构改进及并行 重叠引入	144	6.4 多处理机结构	216
5.1.2 CPU 处理器的工作方式	144	6.4.1 概述	216
5.2 流水线工作方式	146	6.4.2 共享存储型多处理机	217
5.2.1 流水线处理概念和特点	146	6.4.3 分布存储型多计算机结构	221
5.2.2 流水线的分级和分类	148	6.4.4 多处理机发展	222
5.2.3 流水线举例	149	6.5 多处理机高速缓冲存储器(Cache) 一致性	222
5.2.4 流水线性能分析计算	154	6.6 并行处理软件	225
5.2.5 流水线设计技术	162	6.6.1 并行条件	225
5.3 流水线调度	169	6.6.2 程序并行性	226
5.3.1 流水线调度方法	169	6.6.3 并行程序设计语言及其 实现方法	229
		6.6.4 并行处理操作系统	234

6.7 多处理计算机系统举例	238	7.5 多处理机互连网举例	277
6.7.1 MPP大规模并行处理机系统	238	7.5.1 总线结构	277
6.7.2 我国研制的曙光 1000 大规模并行 计算机系统	240	7.5.2 交叉开关网络	279
6.7.3 SMP 共享存储型多处理机	244	7.5.3 多端口存储器	281
6.7.4 机群系统	248	7.5.4 多级互连网	282
习题	252	习题	284
第 7 章 互连网络	255	第 8 章 非冯·诺依曼计算机结构	286
7.1 概述	255	8.1 四种驱动方式计算机模型	286
7.1.1 互连网络概念	255	8.1.1 控制驱动模型	286
7.1.2 互连网络的特性和网络传输 性能参数	255	8.1.2 数据驱动模型	287
7.2 基本互连网函数及表示	258	8.1.3 需求驱动模型	287
7.2.1 互连函数的表示	258	8.1.4 模式匹配驱动	288
7.2.2 几种基本的互连函数	259	8.2 数据流计算机	289
7.3 互连网结构	262	8.2.1 数据流计算机工作原理	289
7.3.1 几种单级互连网	262	8.2.2 数据流程序图和数据流语言	291
7.3.2 基本的循环互连网和多级 互连网	265	8.2.3 数据流计算机的性能分析	296
7.3.3 互连网络的设计特性	268	8.2.4 数据流计算机结构	298
7.3.4 网络拓扑结构	269	8.3 按需求驱动的归约机	303
7.3.5 互联网其他分类	271	8.3.1 函数式程序设计语言	303
7.4 互连网通信模式和消息传递	272	8.3.2 函数式语言的归约机结构	303
7.4.1 互连网通信模式	272	8.4 人工智能计算机	306
7.4.2 互连网中的消息传递机制	273	8.4.1 人工智能计算特征	306
		8.4.2 人工智能计算机分类	306
		习题	307

第1章 概 论

1.1 现代计算机系统

自 1946 年第一台电子数字计算机问世以来, 已经过半个世纪, 经历了五次更新换代, 各代划分的主要标志是硬件和软件技术的明显变革。从所用的器件而言, 由电子管、晶体管到集成电路, 继电器存储, 磁介质存储到集成电路, 乃至大规模和超大规模集成电路。从结构的发展而言, 由单个 CPU 组成的单机系统到流水线机、多处理机到大规模的并行处理机, 从技术的发展上而言, 变址技术、中断技术、微程序技术、缓冲技术共享、重用技术、虚拟技术。从语言及应用方面而言, 由汇编到高级语言, 从单用户到多用户, 到并行处理的语言、编译, 操作系统, 直到如今计算机网络、异构系统处理等等。随着硬、软件的发展, 展示在人们面前的计算机也已大大不同于 50 年前的样子, 而且仍在不停地迅速发展和变化着。

1.1.1 现代计算机的构成

计算机系统不只是最明显的硬件设备, 还应包括与它不可分的系统软件、应用程序、用户接口等形成一个集成的系统, 如图 1-1-1 所示。

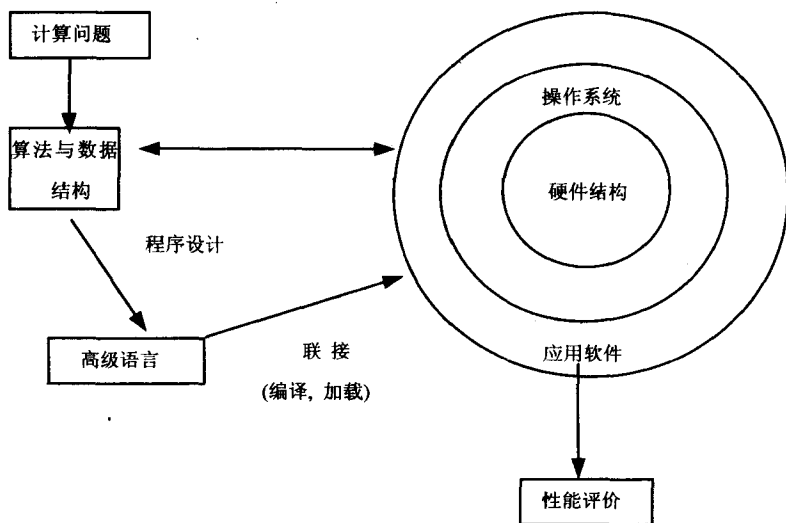


图 1-1-1 现代计算机的构成

(1) 硬件结构 此处的硬件结构是指计算机系统的核心部件。包括中央处理机、存储系统、输入输出接口及系统所具有的各种外部设备。其规模的大小可以差异很大, 小规模计算机硬件, 可以是一个单片机, 一个单处理机。规模大的可以是多处理机, 超级计算机及大规模并行处理机。不论规模大小它在系统中的功能与作用是相同的。

(2) 操作系统 随着系统硬件的发展操作系统必须能支持多处理机、多计算机的并行处理, 能对虚拟存储系统进行管理, 能处理网络通信服务等问题。

(3) 应用软件 用户所编写的程序

- 计算问题是用户需要用计算机解决的问题

- 算法与数据结构 无论是数值计算还是非数值计算，都要通过一定的算法与数据结构把计算问题表示出来，为此算法、数据结构与机器的硬件结构之间应建立映射关系，以便计算机易于处理。

- 高级语言 根据一定算法、数据结构与硬件结构的映射关系，进行高级语言的程序设计，通过编译后加载到硬件系统中去。

(4) 性能评价 对计算机系统性能予以测试及评价，在计算机的研究设计、生产、应用各个阶段都是非常需要的，是计算机系统发展中不可缺少的一部分。在一定的理论、方法与实践的基础上，已形成了比较通用的测试与评价的方法与标准。

1.1.2 计算机系统的层次结构

从计算机系统的研究分析、开发生产与应用来看，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。图 1-1-2 所示的计算机系统可以分为 7 个层次，其中图 1-1-2a、b 表示出了每个层次的组成。

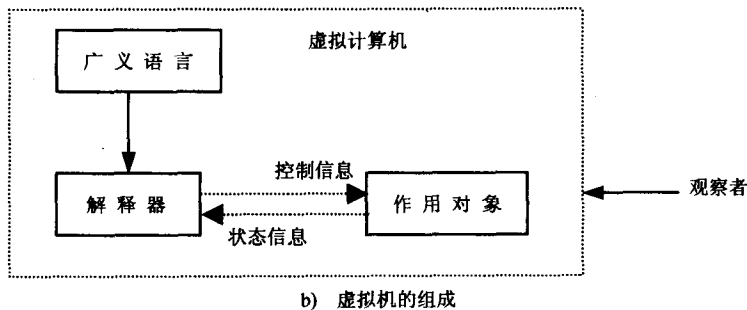
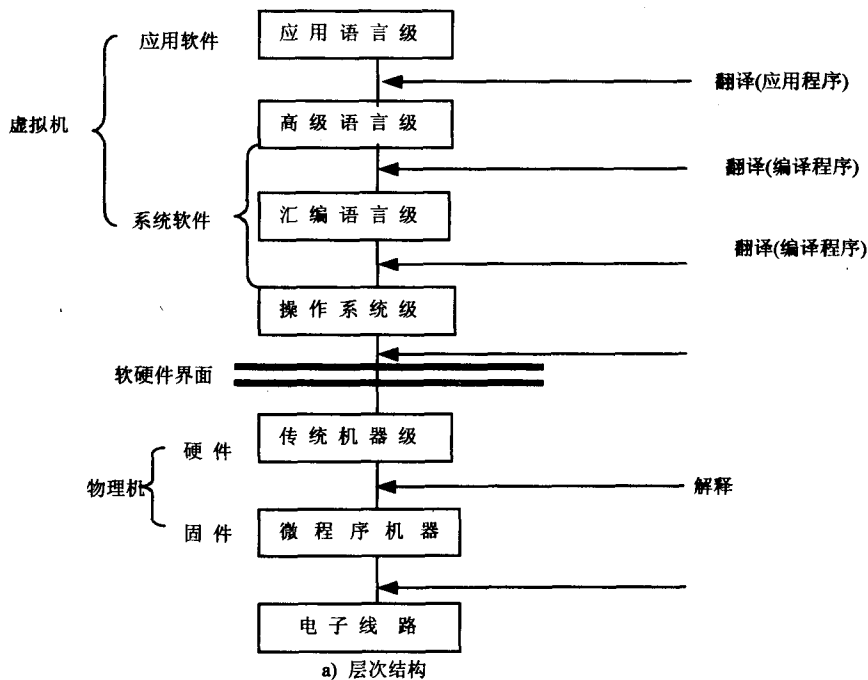


图 1-1-2 计算机系统的层次结构

图中 0 级~2 级为物理机。0 级为硬件实现的逻辑电路,所有的功能都由硬件实现。1 级为微程序级,各种指令由微程序控制执行,微程序控制器由固件实现。2 级为机器语言级,机器语言程序通过解释器形成微程序,由中央处理机运行完成。0~2 这三级的机器及其功能的实现是由硬件或固件完成,因而称为物理机。

3~6 级为虚拟机,其语言的功能均是由软件实现的,由图 1-1-2b 可以看出无论是哪一级均可视为由三部分组成。其中第一部分为作用对象,是同级的程序员与机器的连接部分。程序员用该级的语言所编制的程序,如 5 级则是高级语言的运行程序,4 级则是汇编语言的运行程序,3 级则是操作系统中对各种资源的管理与控制程序。第二部分广义语言,是各级定义的语言。如应用语言级中,针对各种不同的应用领域设计了不同的应用语言,使该领域的专业人员能够很方便地使用这种语言,向计算机表达所要解决的问题。高级语言级,可以是目前所采用的各种高级语言,如 JAVA、C++、VB 等等。4 级为汇编语言;3 级操作系统则是指已定义的各种命令及系统原语。第三部分包括编译器或解释器,通过编译或解释,可以使高一级的语言转换成低一级或者低二级甚至直至机器语言的形式,以便在硬件上运行。

1.2 计算机系统结构

1.2.1 计算机系统结构的基本概念

20 世纪 60 年代初,随着 IBM 360 系统的问世,“计算机系统结构”一词也进入了计算机技术的领域,这标志着计算机技术发展到了一个新的历史时期。“结构”一词来源于建筑学。用来说明系统的外特性,如房屋的建筑,都是由门、窗、墙等组成,但是做为家居、工厂、学校、旅馆等等不同用途的建筑,其外特性是十分不同的。计算机系统结构一词,也是用来说明计算机系统所具有的外特性。

随着计算机的发展,功能要求越来越强,结构越来越复杂,对计算机系统的研究、设计、生产、应用各个方面的任务内容也越来越多、越来越难,逐渐形成了其技术理论的发展方向。计算机系统的设计者所要解决的问题是根据用户的需求以及当前技术发展水平,工艺的水平与条件,决定系统硬件、软件应具有的功能、规模以及所必须采用的关键技术,以使系统既能具有优良的性能,又要保证生产制造的可行性与经济性。1964 年,IBM 公司的系统设计师阿姆达尔(G.M.Amdahl)在介绍 IBM 360 机型时提出了系统结构的概念,他认为“系统结构这个词表示从程序设计者的角度所看到的系统的属性,即概念性结构和功能特性”。但是在此定义中,程序设计者一词比较含糊,没有明确提出是哪一级的程序设计者;此外系统属性的含义也不够明确,究竟指的是哪些性能。所以在此之后的一段时间内又有不少专家学者对系统结构的概念给以补充完善,同时计算机工业的进一步发展,使系统结构的概念逐渐明确了统一了,而且已发展成计算机技术的一个重要的研究方向。

计算机系统结构是指机器语言程序的设计者或是编译程序设计者所看到的计算机系统的概念性结构与功能特性。由计算机的层次结构可以看出,应用级的程序设计者所见到的计算机的性能与传统机的程序设计者一定是不同的,前者用应用语言编程,所以他所看到的机器的功能是能够识别和执行应用语言。而高级语言的程序员在研究、分析、设计、应用计算机时,他所看到的机器都能识别和执行高级语言。所以在他看来 IBM 机与 VAX 机没有什么太大的区别,似乎具有相同的功能特性。然而对汇编一级的程序员来说,IBM 与 VAX 可能在

处理能力性能上相差不大,但这两种机器的功能特性却是非常不同,它们具有不同的数据宽度、不同的指令系统、不同的寻址方式、不同的存储器编址、不同的输入输出方式。即便是对于同一台机器,从不同层次上看,所表现出的功能特性也是不同的。汇编程序员所了解的机器功能、结构,例如上述的指令种类、数据表示、存储器与输入输出的方式,这些硬件所表示出来的属性以及与其他机器的差异,对于高级语言的程序员而言,却看不到,可以认为是不存在的。同样道理,对于汇编级的程序员而言,物理机中所采用的微程序、存储器的缓冲器结构也都是可以不去了解和关心的内容。对于这种本来存在,有差异的事务和属性,从某种角度上看又好像不存在的现象,被称为是“透明性”。透明性的概念在研究开发应用计算机中是十分重要而有效的。

用透明性的概念来分析系统结构的定义,在机器语言或编译程序设计者看来计算机所表现的概念性的结构与功能特性又是指什么呢? Amdahl 所定义的系统结构是指程序员面对的是硬件的系统。也就是指软硬件界面确定后,物理机所表现出的结构与功能的特性。由此可知对系统结构的设计来说,所关心的是如何合理地进行硬、软件功能的分配。计算机中的逻辑功能可以由硬件实现,也可以用软件实现。例如浮点运算功能,在早期的计算机中(如 8080, 2-80, PDP-11)没有浮点运算指令,也没有浮点运算的部件,而是由机器的加法,减法移位等基本的指令编写成子程序,通过调用子程序,完成指定的浮点运算,而当前的机器中都是由硬件来完成的,如 IBM 360 的运算器既可以实现定点运算,也可以组连成浮点运算部件。而 Intel 8086 中,就可以有 8087 协处理器芯片,完成 8 位的浮点运算。目前的 80i86 以上的微处理机,将定点运算和浮点运算组装在一块芯片中。此外如中断级优先权的仲裁、数组的处理、数据格式的转换等,都可以由软件或硬件来完成。可以看出,软硬件可以完成同样的功能,也可以认为软、硬件在逻辑功能上是等价的,或称逻辑功能上的等效性。系统结构则是指在软、硬件的分界面,机器所表现出的外特性。

1.2.2 计算机系统结构、组成与实现

由前面已知计算机系统结构是机器语言级的程序员所了解的计算机的属性即外特性,计算机组成则是指计算机系统结构的逻辑实现。而所谓的计算机的实现是指其计算机组成的物理实现。下面就计算机中的具体内容来说明三者的含义及其相互关系。

(1) 计算机系统结构 可以包含下列几个方面:

① 数据表示: 硬件能够直接识别和处理的数据类型及其表示格式。例如一般的计算机都能识别处理小数、整数、实数、布尔数、字符等。目前的通用机、大型机或功能很强的机器,还可以直接识别处理数组向量、表、阵列等。

② 寄存器定义、数量、使用方式: 汇编程序员在编程时必须了解可编程的寄存器,包括通用寄存器组,变址寄存器,控制用标志寄存器,程序状态字其他专用寄存器。早期的计算机只有少量的寄存器组,如 PDP-11/15,只有 8 个通用寄存器,而 80386 除了有 16 个通用和专用的寄存器之外,还有一组段描述符寄存器部件,在精简指令系统 RISC II 机中,具有 138 个寄存器。

③ 指令系统: 指令系统包括指令的种类,指令格式,各种寻址方式以及指令的控制机构。指令是计算机硬件所能完成的功能。在传统的计算机中,一般而言,功能比较低的机器,指令系统也比较简单,如早期用于培训的 PDP-8 只有 8 条指令,而后来的计算机 PDP11-15 具有近百条的指令而且有功能很强的 12 种寻址方式。

④ 中断系统：包括中断的种类、中断分级、中断处理程序的功能和入口地址。

⑤ 存储系统：各级存储器的编址方式，主存容量，存储系统的结构。

⑥ 输入输出结构：包括 I/O 与 CPU，主存的信息传送方式，设备的访问方式，I/O 的数据传送量及出错指示等。

⑦ 机器工作状态：指目标状态(用户状态)与管理状态(系统状态)的定义与切换。

⑧ 信息保护：指保护方式、硬件对信息保护的支持。

(2) 计算机组成 计算机组成是系统结构的逻辑实现，包括机器内数据流和控制流的组成及逻辑设计。主要解决机器各事件的排序方式与控制机构，各部件的功能及相互间的关系，计算机组成的主要目标是如何提高性能，降低价格。如提高操作的并行度、重叠度，部件的充分应用等。

具体包括以下几方面：

① 数据通路的宽度：数据总线上一次并行传送的信息位数。如 8 位、16 位、32 位等。

② 专用部件的设置：为某种目的设置的专用部件，如提高处理能力设置乘除专用部件，浮点专用部件，字符处理部件，地址运算部件等。

③ 各功能部件，操作部件的并行度和共享程度。如各部件是否串行，重叠，流水，各种不产生冲突的操作是否并行等等。

④ 控制机构的组成方式：指令控制采用硬接或逻辑控制还是采用微程序控制，是否采用多处理机或多功能部件。

⑤ 缓冲技术，排队技术的采用，决定采用哪些缓冲器及各种线性堆栈。

⑥ 可靠性技术的采用和方式的选择。

⑦ 预估、预判、优化处理。

(3) 计算机的实现 计算机逻辑设计之后的物理实现，包括处理机，主存部件的物理结构，器件的集成度，速度的选择，模块、插件、底板的划分与连接，电源、冷却，以及整机的装配技术等。

计算机系统的结构、组成、实现是三个完全不同的概念，相互间有着十分密切的依赖关系和相互的影响。但所要解决的问题，采用的技术和方法又十分不同，例如指令系统中包含哪些指令是结构设计时应决定的问题。而指令的执行，经过取指、取数、执行、送结果四个时间段，以及指令操作的排序是组成应解决的问题，而实现这些逻辑关系采用的器件及具体的电路及模块插件的装配又是实现中应完成的。又如对主存储器的容量与编址方式在结构设计时决定，但在逻辑结构上是否采用多体交叉存储器是计算机组成的任务。由此可看出同一种计算机系统结构可以有多种逻辑组成的方案，而同一种逻辑组成又可以有多种不同的物理实现方法，前者是后者的依据，后者又是前者的基础，一种先进的实现技术将对结构与组成产生巨大的影响。

还应说明的是以上所定义系统结构、组成和实现的概念及其所包含的具体内容是相对的，不是一成不变的。由于计算机技术是正在飞速发展的学科，随着新技术新机种的不断出现，上述的一些概念也相应地有些变化。例如高速缓冲存储器一般做为系统组成的内容，信息的地址转换与替换全部由硬件自动完成，因此对机器程序员来说是透明的。然而有的机器中设置了高速缓存的管理指令，程序员可以参与高速缓存的管理，这就成了系统结构的内容。另外由于超大规模集成电路(VLSI)的迅速发展，集成度越来越高，一块芯片中可以包括有多个

处理器, 就使系统结构的设计内容还要考虑多处理机的问题。除了上述的定义之外, 还应包括功能模块的设计。也就是说计算机系统结构、组成、实现之间的界限随技术的发展也会不断变化。

1.2.3 计算机系统结构的分类

计算机按系统的结构分类通常有三种方法:

(1) 弗林(FLYNN)分类法 在程序执行的过程中, 计算机中只有两种类型的信息在传送, 即控制指令和被处理的数据, 形成了指令流(Instruction Stream)和数据流(Data Stream)。指令流是指处理机在执行程序时控制信息的传送序列; 数据流是指在指令流控制下, 数据信息沿着数据通路传送、加工的序列。由此可见, 指令流和数据流是一个动态的概念, 1966 年 FLYNN(弗林)提出了一种按指令流和数据流的分类法(见图 1-2-1):

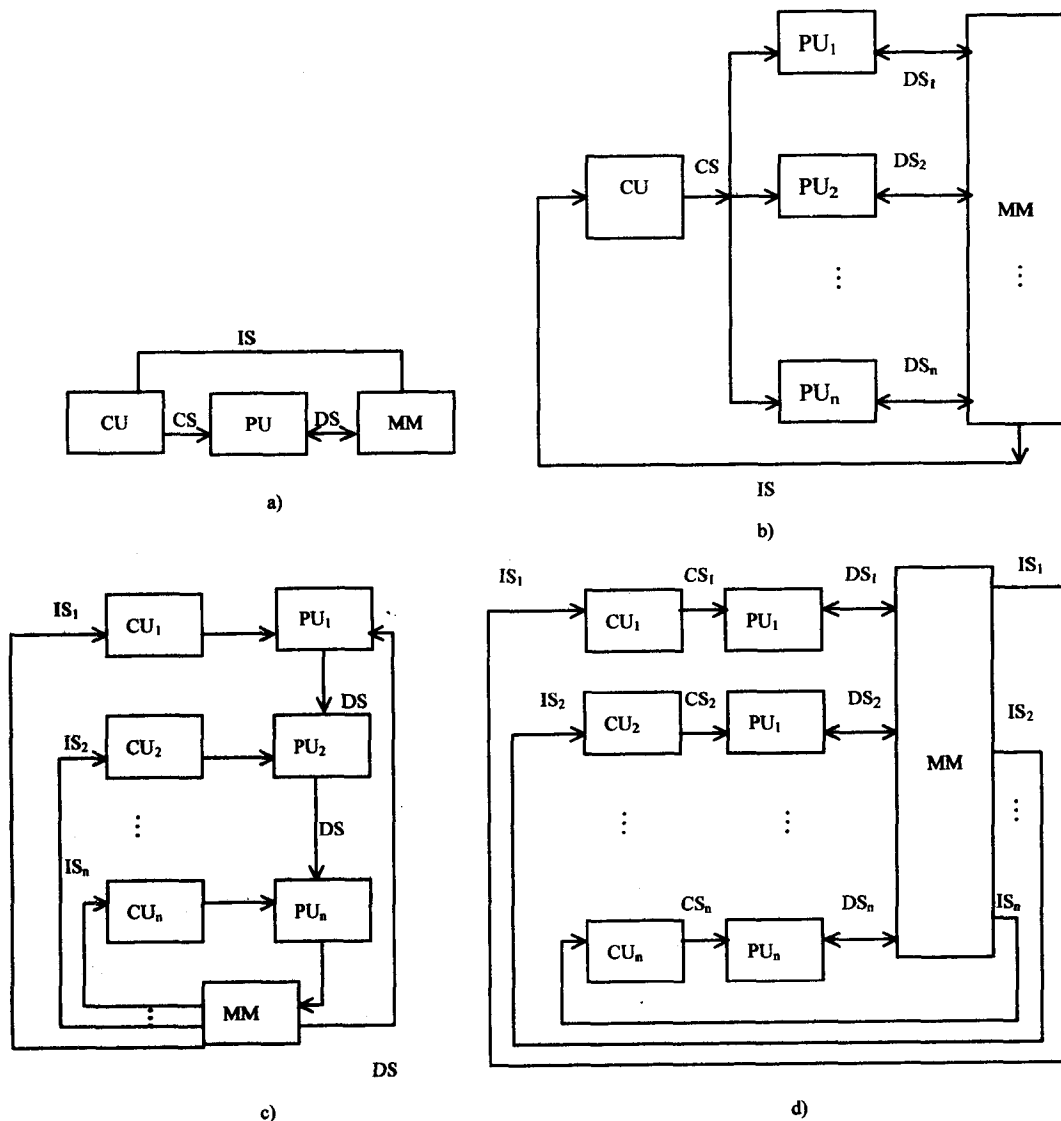


图 1-2-1 FLYNN 分类法

CU—控制部件 PU—处理部件 MM—主存 IS—指令流 CS—控制流 DS—数据流

1) 单指令流、单数据流(SISD-Single Instruction Single Data)结构,即传统的单处理机。它的指令部件一次只对一条指令译码,并只对一个操作部件分配数据。现在通常用的计算机多属此类,如 PDP-11、IBM-360/370、PC 8086、Z-80 等微处理机。

2) 单指令流、多数据流(SIMD-Single Instruction Multiple Data)结构。其典型代表是阵列处理机或并行处理机,它的并行性存在于指令一级。其指令部件根据同一条指令的要求,在同一个控制器的控制下,同时向多个处理部件分配不同的数据。各个数据流的流速应该是相同的,如 ILLIAC IV(64 个单元)和 BELL 实验室的 PEPE(16 单元)都是阵列机。

3) 多指令流、单数据流(MISD)结构,此种形式是多个指令部件,并行发出多个指令流到不同的处理器,对同一个数据流进行处理,目前,与实际机器相对应还有争议,有的把流水线结构,如 RISC 机、向量机归为此类。

4) 多指令流、多数据流(MIMD)结构,是一个多处理机系统,是实现全面并行的理想结构。每个处理器都分别执行系统分配给它的程序。同时执行多个指令流对多个数据流不同的处理,如 IBM 3081/3084、Univac 1100/80, Cray-2 等均属这一类型。

按用途分计算机又可以有专用机与通用机,按规模分又有巨型机、大型机、中型机、小型机、工作站、微型机、单板机、单片机等等,随着技术的发展,大、中、小、巨、微之间难以有个明确的固定的界限。值得说明的是当前经常用到的几种。工作站(Work Station)是集中了微型机与大型机的一些特点构成的一种机型,其性能优越,价格相对便宜,适用于图形处理、网络服务器,复杂的科学计算等;微型机系统是以微处理器为核心组成的系统;当前微型机的结构也有了很大的变化,采用了流水线,多 CPU 的并行处理技术,使性能已接近与某些工作站与中、大型机。另外,单板机与单片机仍有很多的应用。

(2) 冯氏分类法 计算机系统结构发展的一个主要方向是并行性,冯氏法是按最大并行度来分类的。最大并行度是指计算机单位时间内能够处理的最大的二进制位数,设 n 表示一个字中同时处理二进制的位数, m 表示一个位片或功能部件中能同时处理的字数。

1) 字串位串, WSBS, $n=1, m=1$, 指只有一个处理部件,而该部件是串行的每字长为 1 位。

2) 字并位串, WPBS, $n>1, m=1$, 指只有一个处理部件,而该部件的处理字为 n 位,如传统的计算机。如 PDP-11, $n=16, m=1$, IBM 360, $n=32, m=1$ 。

3) 字串位并, WSBP $n=1, m>1$ 指有多个处理部件,但每个处理部件的字长为 1,即串行处理,如大规模的并行处理 MPP 机型 $n=1, m=16384$ 。

4) 字并位并, WPBP $n>1, m>1$ 即多处理部件,各部件的字长也是并行,如 ILLIAC-IV 机具有 64 个处理单元,每个单元的字长为 64 位。

(3) Handler(汉德勒)法 在冯氏分类的基础上,从硬件设备结构的并行级和流水线的程度分类;可以将计算机的硬件设备分成三个层次,程序控制部件的个数,算术逻辑运算部件或处理部件的个数,基本逻辑线路的套数。如 PDP11 机只有一个程序控制部件;一个算术逻辑运算部件;该部件字长为 16 位,具有 64 套电子线路,所以可以表示为 PDP—11=(1, 1, 16)。又如 Cray-1 向量机具有 1 个 CPU,即 1 个程序控制部件;有 12 个条流水线,每条流水线有 8 个功能处理部件;每个部件可处理的字长为 64 位,所以可表示为 Cray—1=(1, 12×8, 64)。

1.3 计算机系统设计

计算机系统结构的设计是整个计算机系统中极为重要的内容,因此也必须遵守计算机系统设计的原则方法和步骤。半个世纪以来,随着计算机系统的研究和生产,对设计的技术也逐渐从定性的分析发展到定量分析。下面介绍的是目前在计算机设计中应该遵循的定量原理。

1.3.1 计算机系统的设计原则

1. 提高计算机速度性能的有效措施是加速那些使用频率高的部件

这是最重要也是被广泛采用的设计准则。道理十分明显,经常使用的部件速度改进后对整机系统的改进影响就大,而很少使用的部件,即使改进后的效果十分明显,但由于很少使用,在整机运行中占极少的部分,因而整机性能不会受多大影响。例如在处理浮点运算时,基本的加、减运算被经常用到,而浮点开平方根的操作则很少用到,所以,系统设计时,就应当着重改进基本的浮点加、减运算。

2. Amdahl 定律

上述论点是定性的分析与设计的原则,但在遇到具体问题时,又如何来衡量使用频度与改进效果的大小呢?如何确定整机系统性能的提高与使用频度与部件改进效果的关系呢?IBM 公司的设计师 Amdahl 在 1967 年提出了一个定量计算的方法,这就是通常所说的 Amdahl 定律。

Amdahl 定律:系统中某一部件由于采用某种改进的执行方式后,整个系统的性能提高了,其衡量指标为加速比。其定义为:

$$\text{加速比} = \frac{\text{采用改进措施后的性能}}{\text{没有采用改进措施的性能}}$$

如果系统性能用执行某任务的时间表示,则可以写为:

$$\text{加速比} = \frac{\text{没有采用改进措施前执行某任务所用时间}}{\text{采用改进措施后执行该任务所用时间}}$$

Amdahl 定律中指明加速比与两个因素有关,一是被改进部分的使用频率的大小,可以用时间参量表示。即某个任务在计算机中执行的总时间中,被改进部分的执行时间所占的百分比的大小;用 f_e 表示,则 f_e 总是小于 1。二是被改进部分本身在改进后所取得的效果,也可以用时间参量来表示,即在执行某任务时,该部件在改进前所需的执行时间与改进后所需的执行时间之比,说明了性能提高的倍数。用 r_e 表示,则 r_e 总大于 1。如果以 S_p 表示加速比, T_e 表示采用改进措施前执行某任务系统所用的时间, T_0 表示采用改进措施后所需的时间,则 Amdahl 定律可表示为:

$$S_p = \frac{T_e}{T_0}$$

在整个 T_0 的时间中改进部分所占的时间只为 $f_e T_e$, 而不可改进的部分为 $T_e(1-f_e)$ 。

$$T_0 = T_e(1 - f_e + \frac{f_e}{r_e})$$

则

$$S_p = \frac{T_e}{T_0} = \frac{1}{(1 - f_e) + \frac{f_e}{r_e}}$$

分析上式可以看出:

当 f_e 很小甚至 $\rightarrow 0$ 时, 则 $S_p \rightarrow 1$ 。说明对这一部分的改进没有意义, 不会影响系统性能。

当 r_e 很大甚至 $\rightarrow \infty$ 时, 则 $S_p = \frac{1}{1 - f_e}$ 。说明无论部件本身改进效果多么明显, 系统的性能受 f_e 的限制。

例: 计算机执行某测试程序, 其中含有大量浮点数据的处理操作, 为提高性能可以采用两种方案, 一是采用硬件实现求浮点数平方根(FPSQR)的操作, 可以使该操作的速度提高 10 倍; 另一种方案是提高所有浮点数据操作(FP)的速度, 使其加快 2 倍。同时已知 FPSQR 操作时间占整个测试程序执行时间的 20%, 而 FP 操作占整个执行时间的 50%, 现比较两种方案。

按两种情况求出其加速比

$$S_p(\text{FPSQR}) = \frac{1}{(1 - 0.2) + \frac{0.2}{10}} = 1.22$$

$$S_p(\text{FP}) = \frac{1}{(1 - 0.5) + \frac{0.5}{2}} = 1.33$$

可见改善所有浮点数据操作的方案更好一些。

3. 程序访问局部性原理

程序访问局部性原理说明了计算机在程序执行过程中呈现出的一种规律。就是程序往往重复使用它刚刚使用过的数据和指令。实际统计表明一个程序执行中的 90% 的时间往往是花费在 10% 的程序代码上。通过测试, 三个典型的程序中经常使用的代码比例分别是:

Gcc(Gnu·C 编译) 13%

Spice (CAD 电路分析软件) 9.5%

TeX(文本处理软件) 9.3%

对此种规律称为程序访问局部性。它反映在两方面的局部性, 即时间和空间。所谓时间局部性是指近期被访问的代码, 很可能不久又将再次被访问; 空间局部性是指地址上相邻近的代码可能会被连续地访问。

程序局部性原理是计算机系统设计很重要的指导原则, 如目前存储系统层次结构的设计就是以此为理论基础。

1.3.2 计算机系统结构设计

1. 硬件功能分配的基本原则

计算机系统设计中最重要的问题，也是首先遇到的问题就是系统结构的设计，而系统结构设计的主要任务就是软、硬件功能的分配，也就是系统中软、硬件的界面划分。在进行软、硬件功能分配时，设计者的主要依据应该是整个系统所确定的“计算机系统的功能要求、性能要求、成本要求。”

• 功能要求应该包括以下几个方面：

首先是计算机应用领域：例如，是专用机还是通用机，如果是通用机，又需要了解是用于科学计算还是事务处理，是面向个人家用，还是科研、行政机构用。如果是专用，就应该了解专用于某个方面，因为根据应用领域就可以得知其他的功能要求及特点。其次是对软件兼容性的要求，操作系统的要求，以及各种通用国际标准的要求。如总线标准，网络标准，各种接口标准等。

• 计算机性能要求，包含很多因素，如速度，容量，输入/输出的种类，可靠性、可维护性、安全性、容错能力，人机交互的能力等等。

• 计算机的成本要求 计算机硬件和软件的功能分配在相当大的范围内是可以变化的，也就是说对某些功能而言，可以由硬件实现也可以由软件实现或者是固件实现。系统结构的设计者应该如何确定呢？

一般说来，计算机硬件实现的功能速度比较高，对内存容量的消耗比较小。但是计算机的灵活性和适应性比较差，增加了硬件的成本。而软件实现的功能则正巧相反，灵活性，适应性好，降低了硬件成本，但是速度比较慢，增加了对内存存储容量的要求。根据以上所述如何取舍呢？主要考虑的标准应该是性能价格比。

设计者首先考虑的是要保证用户或者是市场对功能和性能的要求，例如对一台面向科学计算或面向实时控制的计算机，则必须采取措施满足其速度的要求，如多采用硬件实现某些功能，甚至采用多功能部件，或并行处理部件等，其次要考虑成本，成本可分为开发成本和复制成本，对于开发成本，软、硬件的投入都比较大，都是不可忽视的。从复制成本看，硬件却大大超过软件，当产量大，复制的数据多时，这一点就越明显，硬件的复制需要提供同样产品及生产的投入，而软件只需复制一份磁盘或光盘就可以了。设计者在确定软、硬件界面时必须对性能与价格两方面统一考虑，以获得最优的设计。

值得指出的是随着技术的发展，上述的观念也在不断变化，硬件由于集成技术的发展，成本不断降低，因而用硬件固件实现的功能越来越多。软件在初期是不计成本，免费供给的，随软件技术的发展，软件的成本不断提高，所以系统结构面临的软、硬件界面的划分也在不断变化。表 1-3-1 给出了计算机发展以来，软、硬件功能分配的状况。

2. 系统结构设计的方法

基于计算机系统层次结构的基础上，其设计方法可以有以下三种：

• 自下而上的方法：基于硬件技术所具有的条件，设计工作起始于最低层的元器件、组件的选择及逻辑设计，完成传统机器的研制。在此硬件基础上，再向上逐级进行软件的开发。这种方法是计算机发展的早期采用的，因为当时硬件成本高，技术水平低，软件设计应该依附于硬件，现在已基本不用了。

• 自上而下的方法：这种方法首先考虑的是如何满足用户的要求，即先确定应用级虚拟机所具有的基本功能特性，如基本命令，应用语言的结构，格式，数据的类型等。然后逐级