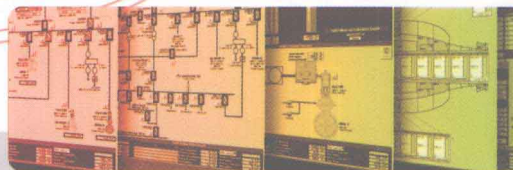
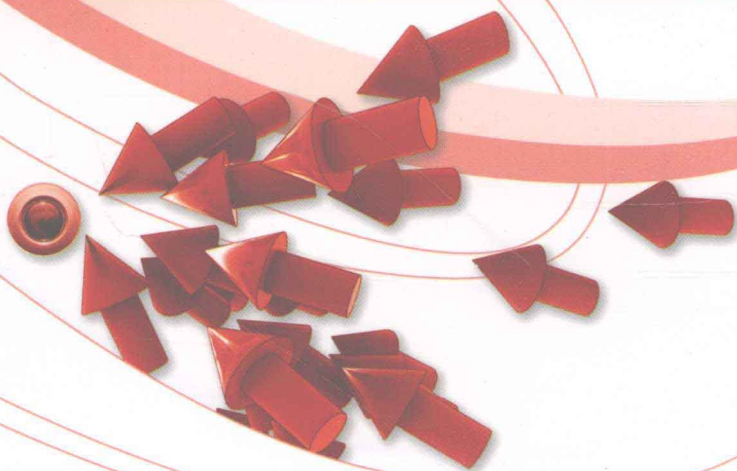




普通高等教育“十二五”规划教材



工程创新型“十二五”规划计算机精品教材

计算机系统结构

■ 李学干 编著

电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

普通高等教育“十二五”规划教材
工程创新型“十二五”规划计算机精品教材

计算机系统结构

李学干 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统地讲述计算机系统结构的基本概念、基本原理、基本结构、基本分析方法以及近年来在该领域的进展。

全书共 8 章。主要内容包括:计算机系统结构基础知识及并行性开发;数据表示、寻址方式与指令系统的设计、优化、发展和改进;存储、中断、总线与输入/输出系统;虚拟存储器、高速缓冲存储器、三级存储层次和主存保护;重叠方式和流水方式的标量处理机以及指令级高度并行的超级处理机;向量的流水处理和向量流水处理机、阵列处理机的原理、并行算法和互连网络;多处理机的硬件结构、多 Cache 的一致性、程序的并行性、性能、操作系统和多处理机的发展;数据流计算机和归约机。

本书内容丰富,取材适当,每章均有大量案例和习题。书末附有各章内容提要、知识点、重点和难点以及各章习题的解答。本书可作为计算机专业本科生和相关专业研究生的教材,也可作为相关领域科技人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

计算机系统结构/李学干编著. —北京:电子工业出版社, 2011. 12

工程创新型“十二五”规划计算机精品教材

ISBN 978 - 7 - 121 - 14948 - 1

I. ① 计… II. ① 李… III. ① 计算机体系结构 - 高等学校 - 教材 IV. ① TP303

中国版本图书馆 CIP 数据核字(2011)第 221967 号

责任编辑:韩同平 特约编辑:林宏峰

印 刷: 北京市李史山胶印厂

装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787 × 1092 1/16 印张:17.75 字数:480 千字

印 次:2011 年 12 月第 1 次印刷

印 数:3 000 册 定价:36.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

本书是从计算机组织和结构的角度来引导学生深入学习和领会计算机系统的。整个计算机系统是一个由软件和硬件组合的复杂综合体。随着计算机硬件在功能、性能、集成度、可靠性、价格上的不断改进,计算机软件日趋复杂以及应用要求的不断拓宽和深化,需要我们研究如何更好、更合理地分配计算机系统软件和硬件的功能,同时研究应如何更好地实现分配给硬件的那部分功能,让系统有尽可能高的性能价格比。这对从事计算机系统结构设计、硬件设计、系统软件开发和高层次应用系统开发的人来说都是应该了解和掌握的知识。通过学习,希望能进一步加深理解计算机系统的整体概念,正确掌握计算机系统结构的基本概念、基本原理,了解比较成熟的基本结构,学会正确进行结构设计的思想和方法,提高分析问题和解决问题的能力,了解近十几年里,在并行处理和系统结构领域里的重要进展。

全书共 8 章。第 1 章是计算机系统结构基础知识及系统结构中的并行性开发,讲述计算机系统的多级层次结构,计算机系统结构、组成、实现的定义和相互关系,计算机系统的软硬取舍原则、性能评测及定量设计原理,软件、应用、器件的发展对系统结构的影响,系统结构中的并行性开发和计算机系统的分类;第 2 章讲述数据表示,寻址方式,指令系统的设计、优化、发展和改进;第 3 章讲述存储系统的基本要求和并行主存系统,中断的分类、分级和中断系统的软硬件功能分配,总线的分类、控制技术、通信技术、数据宽度和总线线数,输入/输出系统的通道处理机的工作原理、流量设计和外围处理机;第 4 章讲述存储体系概念,虚拟存储器、Cache 存储器的构成、地址映像、地址变换、替换算法、实现中的问题和性能分析,三级存储体系的三种形式及存储系统的保护;第 5 章讲述重叠方式和流水方式的标量处理机的原理、相关控制及指令级高度并行的超级处理机(超标量处理机、超长指令字处理机、超流水线处理机、超标量超流水线处理机);第 6 章讲述向量的流水处理,向量流水处理机和阵列处理机的原理、并行算法、互连网络和共享主存构形的阵列处理机中并行存储器的无冲突访问,脉动流水处理机的原理和通用脉动阵列结构;第 7 章讲述多处理机的概念、问题和硬件结构、紧耦合多处理机多 Cache 的一致性问题和解决办法、程序的并行性和性能、操作系统和多处理机的发展(分布式共享主存储器多处理机、对称多处理机、多向量多处理机、并行向量机、大规模并行处理机 MPP、机群系统);第 8 章讲述数据流机和归约机。

本书相应课程“计算机系统结构”应在“数字逻辑”、“计算机组成原理”、“微型机原理和接口技术”、“汇编语言”、“高级语言程序设计”等课程之后开设,学生最好有“数据结构”方面的知识。本课程也可以在“操作系统”、“编译原理”等课程之后或与它们同时开设。“计算机系统结构”的参考教学课时数为 64~72 学时,可以视情况适当增减。

本书内容丰富,取材适当,重点突出,难点分散,文字通俗易懂,各章均有大量的例题和习题。

为了让读者了解各章的内容及知识点要求,附录 A 给出了每章“提要”、“知识点”、“重点”和“难点”,对知识点提出了能力层次的要求。

附录 B 给出了各章习题的解答。这些均对教师教学和学生自己检验所学知识达到的程

度提供帮助。

本书是计算机专业本科生或相关专业研究生的教材,也可作为计算机科学工作者的参考书。

本书由西安电子科技大学李学干教授编写,参加本书编写的还有朱兰娣、李海燕、王哲。电子工业出版社为本书编辑出版做了大量工作,在此表示衷心的感谢。

由于作者水平有限,书中难免有错漏之处,恳请读者给予批评指正。

编著者

目 录

第 1 章 计算机系统结构的基础知识及并行性开发	(1)
1.1 计算机系统的层次结构	(1)
1.2 计算机系统结构、计算机组成和计算机实现	(2)
1.2.1 计算机系统结构的定义和内涵	(2)
1.2.2 计算机组成和计算机实现的定义和内涵	(3)
1.2.3 计算机系统结构、组成、实现的相互关系与影响	(5)
1.3 计算机系统的软硬件取舍、性能评测及定量设计原理	(7)
1.3.1 软、硬件取舍的基本原则	(7)
1.3.2 计算机系统的性能评测及定量设计原理	(8)
1.3.3 计算机系统设计的主要任务和方法	(13)
1.4 软件、应用、器件的发展对系统结构的影响	(15)
1.4.1 软件发展对系统结构的影响	(15)
1.4.2 应用发展对系统结构的影响	(20)
1.4.3 器件发展对系统结构的影响	(22)
1.5 系统结构中的并行性开发及计算机系统的分类	(23)
1.5.1 并行性的概念与开发	(23)
1.5.2 计算机系统的分类	(26)
习题 1	(28)
第 2 章 数据表示、寻址方式与指令系统	(30)
2.1 数据表示	(30)
2.1.1 数据表示与数据结构	(30)
2.1.2 高级数据表示	(31)
2.1.3 引入数据表示的原则	(36)
2.1.4 浮点数尾数基值大小和下溢处理方法的选择	(36)
2.2 寻址方式	(43)
2.2.1 寻址方式的 3 种面向	(43)
2.2.2 寻址方式在指令中的指明	(44)
2.2.3 程序在主存中的定位技术	(44)
2.2.4 物理主存中信息的存储分布	(46)
2.3 指令系统的设计和优化	(47)
2.3.1 指令系统设计的基本原则	(47)
2.3.2 指令操作码的优化	(48)
2.3.3 指令字格式的优化	(51)
2.4 指令系统的发展和改进	(55)

2.4.1	两种途径和方向(CISC 和 RISC)	(55)
2.4.2	按 CISC 方向发展和改进指令系统	(55)
2.4.3	按 RISC 方向发展和改进指令系统	(60)
习题 2		(66)
第 3 章	存储、中断、总线与 I/O 系统	(68)
3.1	存储系统的基本要求和并行主存系统	(68)
3.1.1	存储系统的基本要求	(68)
3.1.2	并行主存系统	(69)
3.2	中断系统	(72)
3.2.1	中断的分类和分级	(72)
3.2.2	中断的响应次序与处理次序	(74)
3.2.3	中断系统的软、硬件功能分配	(76)
3.3	总线系统	(77)
3.3.1	总线的分类	(77)
3.3.2	总线的控制方式	(78)
3.3.3	总线的通信技术	(80)
3.3.4	数据宽度与总线线数	(82)
3.4	输入/输出系统	(83)
3.4.1	输入/输出系统概述	(83)
3.4.2	通道处理机的工作流量和流量设计	(84)
3.4.3	外围处理机	(90)
习题 3		(91)
第 4 章	存储体系	(93)
4.1	基本概念	(93)
4.1.1	存储体系及其分支	(93)
4.1.2	存储体系的构成依据	(94)
4.1.3	存储体系的性能参数	(94)
4.2	虚拟存储器	(95)
4.2.1	虚拟存储器的管理方式	(95)
4.2.2	页式虚拟存储器的构成	(100)
4.2.3	页式虚拟存储器实现中的问题	(109)
4.3	高速缓冲存储器	(113)
4.3.1	工作原理和基本结构	(113)
4.3.2	地址的映像与变换	(115)
4.3.3	Cache 存储器的 LRU 替换算法的硬件实现	(121)
4.3.4	Cache 存储器的透明性及性能分析	(122)
4.4	三级存储体系	(125)
4.5	存储系统的保护	(127)
习题 4		(130)
第 5 章	标量处理机	(134)

5.1	重叠方式	(134)
5.1.1	重叠方式与一次重叠	(134)
5.1.2	相关处理	(136)
5.2	流水方式	(141)
5.2.1	基本概念	(141)
5.2.2	标量流水线的主要性能	(145)
5.2.3	标量流水线的相关处理和控制机构	(149)
5.3	指令级高度并行的超级处理机	(159)
5.3.1	超标量处理机	(160)
5.3.2	超长指令字处理机	(161)
5.3.3	超流水线处理机	(162)
5.3.4	超标量超流水线处理机	(162)
	习题5	(162)
第6章	向量处理机	(166)
6.1	向量的流水处理与向量流水处理机	(166)
6.1.1	向量的处理和向量的流水处理	(166)
6.1.2	向量流水处理机的结构举例	(167)
6.1.3	通过并行、链接提高性能	(168)
6.1.4	提高向量流水处理速度的其他办法	(170)
6.2	阵列处理机的原理	(173)
6.2.1	阵列处理机的构形和特点	(173)
6.2.2	ILLIAC IV的处理单元阵列结构	(174)
6.2.3	ILLIAC IV的并行算法举例	(175)
6.3	SIMD 计算机的互连网络	(178)
6.3.1	互连网络的设计目标与互连函数	(178)
6.3.2	互连网络应抉择的几个问题	(179)
6.3.3	基本的单级互连网络	(180)
6.3.4	基本的多级互连网络	(182)
6.3.5	全排列网络	(188)
6.4	共享主存构形的阵列处理机中并行存储器的无冲突访问	(189)
6.5	脉动阵列流水处理机	(191)
6.5.1	脉动阵列结构的原理	(191)
6.5.2	通用脉动阵列结构	(193)
	习题6	(194)
第7章	多处理机	(197)
7.1	多处理机的概念、问题和硬件结构	(197)
7.1.1	多处理机的基本概念和要解决的技术问题	(197)
7.1.2	多处理机的硬件结构	(198)
7.2	紧耦合多处理机多 Cache 的一致性问题的	(207)
7.3	多处理机的并行性和性能	(209)

7.3.1 并行算法	(209)
7.3.2 程序并行性的分析	(211)
7.3.3 并行语言与并行编译	(213)
7.3.4 多处理机的性能	(217)
7.4 多处理机的操作系统	(218)
7.5 多处理机的发展	(220)
习题7	(224)
第8章 数据流计算机和归约机	(226)
8.1 数据流计算机	(226)
8.1.1 数据驱动的概念	(226)
8.1.2 数据流程图和语言	(228)
8.1.3 数据流计算机的结构	(232)
8.1.4 数据流计算机存在的问题	(232)
8.1.5 数据流计算机的进展	(233)
8.2 归约机	(234)
习题8	(236)
附录A 各章内容提要、知识点、重点和难点	(237)
附录B 各章习题解答	(244)
参考文献	(275)

第 1 章 计算机系统结构的基础知识及并行性开发

1.1 计算机系统的层次结构

从使用语言的角度,一台由软、硬件组成的通用计算机系统可以被看成按功能划分的由多层机器级组成的层次结构。层次结构由高到低依次为应用语言机器级、高级语言机器级、汇编语言机器级、操作系统机器级、传统机器语言机器级和微程序机器级,如图 1-1 所示。具体的计算机系统,层次数多少可以不同。

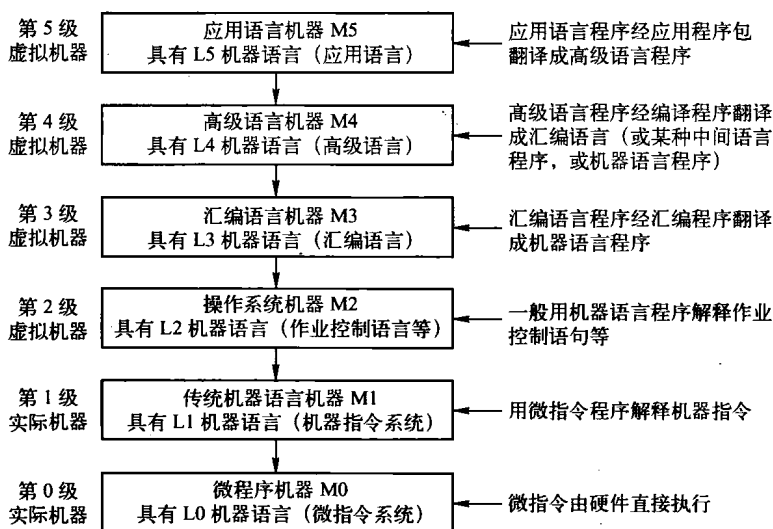


图 1-1 计算机系统的多级层次结构

对使用某一级语言编程的程序员来讲,只要熟悉和遵守该级语言的使用规定,所编程序总能在该机器上运行并得到结果,而不用考虑这个机器级是如何实现的。就好像该程序员有了一台可以直接使用这种语言作为机器语言的机器一样。这里,“机器”被定义为能存储和执行相应语言程序的算法和数据结构的集合体。实际上,只有二进制机器指令(即传统所讲的机器语言)与机器硬件直接对应,方可直接被硬件识别和执行。

各机器级的实现采用翻译技术或解释技术,或者这两种技术的结合。翻译(Translation)技术是先用转换程序将高级机器级上的程序整个地变换成低级机器级上等价的程序,然后再在低级机器级上实现的技术。解释(Interpretation)技术则是在低级机器级上用它的串语句或指令来仿真高级机器级上的一条语句或指令的功能,是通过对高级的机器级语言程序中的每条语句或指令逐条解释来实现的技术。

应用语言虚拟机器级 M5 是为满足专门的应用设计的。使用面向某种应用的应用语言(L5)编写的程序一般是通过应用程序包翻译成高级语言(L4)程序后,再逐级向下实现的。

高级语言机器级 M4 上的程序可以先用编译程序整个翻译成汇编语言(L3)程序或机器语言(L1)程序,再逐级或越级向下实现,也可以用汇编语言(L3)程序、机器语言(L1)程序,甚至微指令语言(L0)程序解释实现。汇编语言(L3)源程序则是先用汇编程序整个将它转换成等效的二进制机器语言(L1)目标程序,再在传统机器级 M1 上实现。操作系统程序虽然已经发展成用高级语言(如 C 语言)编写,但最终还是要用机器语言程序或微指令程序来解释。它提供了传统机器级 M1 所没有,但为汇编语言和高级语言使用和实现所用的基本操作、命令及数据结构,例如,文件管理、存储管理、进程管理、多道程序共行、多重处理、作业控制等所用到的操作命令、语句和数据结构等。因此,操作系统机器级 M2 放在传统机器级 M1 和汇编语言机器级 M3 之间是适宜的。传统机器级采用组合逻辑电路控制,其指令可直接用硬件来实现,也可以采用微程序控制,用微指令(L0)程序来解释实现。微指令直接控制硬件电路的动作。

就目前的状况来看,M0 用硬件实现,M1 用微程序(固件)实现,M2 ~ M5 大多用软件实现。所谓固件(Firmware),是一种具有软件功能的硬件,例如将软件固化在只读存储器这种大规模集成电路的硬件、器件上,就是一种固件。以软件为主实现的机器称为虚拟机器,以区别于由硬件或固件实现的实际机器。虚拟机器不一定全由软件实现,有些操作也可用固件或硬件实现。例如操作系统的某些命令可用微程序或硬件实现。

1.2 计算机系统结构、计算机组成和计算机实现

1.2.1 计算机系统结构的定义和内涵

从计算机的层次结构角度来看,系统结构(System Architecture)是对计算机系统中各级界面的定义及其上下的功能分配。每级都有其自己的系统结构。在此,先说明有关“透明”的概念。客观存在的事物或属性从某个角度看不到,简称对它是透明(Transparent)的。不同机器级程序员所看到的计算机属性是不同的,它就是计算机系统不同层次的界面。系统结构就是要研究对于某级,哪些属性应透明,哪些属性不应透明。透明可简化该级的设计,但因无法控制,也会带来不利影响。因此,要正确进行透明性取舍。

计算机系统结构也称计算机系统的体系结构(Computer Architecture),它只是系统结构中的一部分,指的是传统机器级的系统结构。其界面之上包括操作系统级、汇编语言级、高级语言级和应用语言级中所有软件的功能,该界面之下包括所有硬件和固件的功能。因此,它是软件和硬件/固件的交界面,是机器语言、汇编语言程序设计者,或编译程序设计者看到的机器物理系统的抽象。

结论:计算机系统结构研究的是软、硬件之间的功能分配以及对传统机器级界面的确定,提供机器语言、汇编语言程序设计者或编译程序生成系统为使其设计或生成的程序能在机器上正确运行应看到和遵循的计算机属性。

就目前的通用机来说,计算机系统结构的属性包括:

- (1) 硬件能直接识别和处理的数据类型及格式等的表示。
- (2) 最小可寻址单位、寻址种类、地址计算等的寻址方式。
- (3) 通用/专用寄存器的设置、数量、字长、使用约定等的寄存器组织。
- (4) 二进制或汇编指令的操作类型、格式、排序方式、控制机构等的指令系统。

- (5) 主存的最小编址单位、编址方式、容量、最大可编址空间等的存储系统组织。
- (6) 中断的分类与分级、中断处理程序功能及入口地址等的中断机构。
- (7) 系统机器级的管态和用户态的定义与切换。
- (8) 输入/输出设备的连接、使用方式、流量、操作结束、出错指示等的机器级 I/O 结构。
- (9) 系统各部分的信息保护方式和保护机构等属性。

【例 1-1】 IBM PC 系列和 VAX - 11 系列的指令系统、寻址方式、寄存器组织、I/O 设备连接方式等都不一样,从传统机器语言程序员或汇编语言程序员看,概念性结构和功能特性差异很大。要使他们所编的程序能运行,应了解的计算机属性大不相同,但高级语言程序员却看不到。

1.2.2 计算机组成和计算机实现的定义和内涵

从计算机系统结构的内涵,可以看出,机器级内部的数据流和控制流的组成、逻辑设计和器件设计等都不属于计算机系统结构,就是说,对计算机系统结构设计是透明的。它们属于计算机组成或计算机实现的范畴。

1. 计算机组成

计算机组成(Computer Organization)指的是计算机系统结构的逻辑实现,包括机器级内部的数据流和控制流的组成以及逻辑设计等。

计算机组成着眼于机器级内部各事件的排序方式与控制机构、各部件的功能及各部件间的联系。它要解决的问题是在所希望达到的性能和价格情况下,怎样更好、更合理地把各种设备和部件组织成计算机,来实现所确定的系统结构。20 世纪 60 年代以来,计算机组成设计主要是围绕提高速度,着重从提高操作的并行度、重叠度,以及功能的分散和设置专用功能部件来进行的。

计算机组成设计要确定的方面一般应包括:

- (1) 数据通路宽度(数据总线一次并行传送的信息位数)。
- (2) 专用部件的设置(是否设置乘除法、浮点运算、字符处理、地址运算等专用部件,设置的数量与机器要达到的速度、价格及专用部件的使用频率等有关)。
- (3) 各种操作对部件的共享程度(分时共享使用程度高,虽限制了速度,但价格便宜。设置部件多降低共享程度,因操作并行度提高,可提高速度,但价格也会提高)。
- (4) 功能部件的并行度(是用顺序串行,还是用重叠、流水或分布式控制和处理)。
- (5) 控制机构的组成方式(用硬件还是微程序控制,是单机处理还是多机或功能分布处理)。
- (6) 缓冲和排队技术(部件间如何设置及设置多大容量的缓冲器来协调它们的速度差。用随机、先进先出、先进后出、优先级,还是用循环方式来安排事件处理的顺序)。
- (7) 预估、预判技术(为优化性能用什么原则预测未来行为)。
- (8) 可靠性技术(用何种冗余和容错技术来提高可靠性)。

2. 计算机实现

计算机实现(Computer Implementation)指的是计算机组成的物理实现,包括处理机、主存等部件的物理结构,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。

计算机实现的设计着眼于器件技术和微组装技术,其中,器件技术起着主导作用。

【例 1-2】 指令系统的确定属于计算机系统结构。指令的实现,如取指令、指令操作码译码、计算操作数地址、取数、运算、送结果等的操作安排和排序属于计算机组成。实现这些指令功能的具体电路、器件的设计及装配技术属于计算机实现。

确定指令系统中是否要设乘法指令属于计算机系统结构。乘法指令是用专门的高速乘法器实现,还是靠加法器和移位器经一连串时序信号控制其相加和右移来实现属于计算机组成。乘法器、加法-移位器的物理实现,如器件的类型、集成度、数量、价格,微组装技术的确定和选择属于计算机实现。

主存容量与编址方式(按位、按字节还是按字访问等)的确定属于计算机系统结构。为达到性能价格要求,主存速度应该为多少,逻辑结构是否采用多体交叉属于计算机组成。主存器件的选定,逻辑设计,微组装技术的使用属于计算机实现。

【例 1-3】 IBM 370 系列有 115,125,135,145,158,168 等由低档到高档的多种型号机器。从汇编语言、机器语言程序设计者看的概念性结构都如图 1-2 所示。它们均是由中央处理机/主存—通道—设备控制器—外设 4 级构成的,以主存为中心,采用通道方式输入/输出。从层次结构看,IBM 370 系列中不同型号的机器从高级语言机器级、汇编语言机器级到传统机器语言机器级都是相同的,只是使用不同的组成和实现,不同的微程序机器级,使机器性能、价格不同。因此,设计何种系列机属于计算机系统结构,而系列内不同型号计算机的组织属于计算机组成。

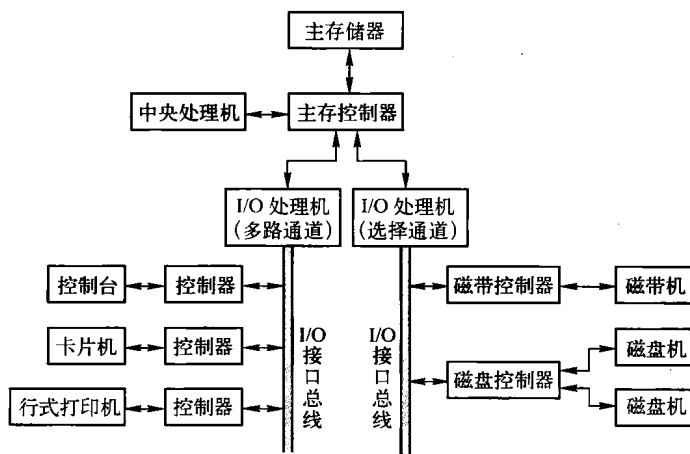


图 1-2 IBM 370 系列的概念性结构

IBM 370 系列的中央处理机都有相同的机器指令和汇编指令系统,只是指令的分析、执行在低档机上为顺序进行方式,在高档机上则采用重叠、流水或其他并行处理方式。程序设计者编程时所看到的数据形式(即数据表示)都是相同的 32 位字长,定点数都是半字长 16 位或全字长 32 位,浮点数都是单字长 32 位、双字长 64 位或四字长 128 位,如图 1-3(a)所示。由于速度、价格的要求不同,在组成和实现时,数据通路宽度(数据总线线数)可以分别采用 8 位、16 位、32 位或 64 位,如图 1-3(b)所示。一个 64 位的字在 8 位数据通路宽度的机器上需分 8 次传送完,而在 64 位数据通路宽度的机器上却只需一次即可传送完,速度快了,但硬件多了,价格贵。因此,数据总线宽度对程序员是透明的,是不需要知道的。

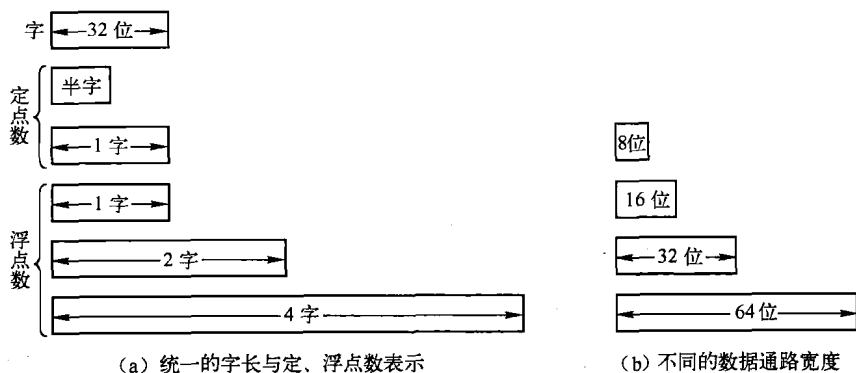


图 1-3 IBM 370 系列字长、数的表示和数据通路宽度

IBM 370 系列的各档机器都采用通道方式进行输入/输出,但在计算机组成上,低档机器可以采用结合型通道,让通道的功能借用中央处理机的某些部件完成。同一套硬件分时执行中央处理机和通道的功能,虽然系统速度性能低,但可以降低成本。而高档机器上却采用独立型通道,单独设置通道硬件,与中央处理机并行,成本虽高,但系统速度提高了。

结论:机器/汇编指令系统、数据表示、是否采用通道方式输入/输出的确定属于系统结构;指令采用顺序、重叠、流水还是其他方式解释,数据通路宽度的确定,通道采用结合型还是独立型,均属于计算机组成。

【例 1-4】 DEC 公司早先推出的 PDP-11 系列是以单总线结构著称的,它不属于计算机系统结构。为适应不同速度和价格的要求,不同型号机器仍使用了多种不同的总线。但是,它们都具有相同的 I/O 连接和使用方式,即将 I/O 设备端口寄存器在逻辑上看做是主存的一部分,与主存统一编址,通过访问主存的这些指定单元来实现与指定的 I/O 设备通信,完成对该设备的读/写等。

因此对 PDP-11 或后来的 VAX-11 来说,单总线结构属于计算机组成,其机器级的 I/O 连接和使用方式才属于计算机系统结构,是程序设计者编写 I/O 程序时应当看到的。

1.2.3 计算机系统结构、组成、实现的相互关系与影响

计算机系统结构、组成、实现三者互不相同,但又相互影响。从前面的例子可以看出,相同结构(如指令系统相同)的计算机,可以因速度不同而采用不同的组成。例如,指令间既可以顺序执行,也可以重叠执行以提高性能。乘法指令既可以用专门的乘法器实现,也可以用加法器、移位器等经重复加、移位来实现,这取决于性能、价格、乘法指令使用频度及所用乘法的运算方法。高速高频的可用专门乘法器,否则宜用后一种方法来降低价格。

同样,一种组成可有多种不同的实现方法。如主存器件可用双极型的,也可用 MOS 型的;可用 VLSI 单片,也可用多片小规模集成电路组成。这取决于要求的性能价格比及器件技术状况。

结构不同会使可能采用的组成技术不同。

【例 1-5】 为了实现

$$A: = B + C$$

$$D: = E * F$$

若采用面向寄存器的系统结构,其程序可以是

```
LOAD R1,B
ADD R1,C
STORE A,R1
LOAD R2,E
MPY R2,F
STORE D,R2
```

而对于面向主存的三地址寻址方式的结构,其程序可以是

```
ADD B,C,A
MPY E,F,D
```

要提高运算速度,可让相加与相乘并行,为此这两种结构在组成上都要求设置独立的加法和乘法器。但对于面向寄存器的结构还要求 R1 和 R2 能同时被访问,而对于面向主存的三地址寻址的结构并无此要求,但是要求能同时形成多个访存操作数地址和能同时访存。

组成也会影响结构。微程序控制就是一个范例。通过改变控制存储器中的微程序,就可改变系统的机器指令,改变结构。在一台计算机上提供对应多种指令系统的微程序,动态切换,结构可变,这是硬联控制组成技术无法做到的。另外,增加多倍长运算、十进制运算、字符行处理、矩阵乘、多项式求值、三角函数求值、查表、字节测试、开方等复合机器指令和宏指令,由微程序解释实现,因为减少了大量访主存取机器指令的次数,速度比用基本机器指令构成的机器语言子程序实现要快几倍到十几倍。如果没有组成技术的进步,结构的进展是不可能的。

因此,系统结构的设计必须结合应用考虑,为软件和算法的实现提供更多更好的支持,同时要考虑可能采用和准备采用的组成技术。结构设计应避免过多地或不合理地限制各种组成、实现技术的采用与发展,尽量做到既能方便地在低档机上用简单、便宜的组成实现,又能在高档机上用复杂、较贵的组成实现,使它们都能充分发挥出实现方法所带来的好处,这样,结构才有生命力。

组成设计向上决定于结构,向下受限于实现技术。然而,它们是可以与实现折中权衡的。

例如,为达到所要求的速度,可用较简单的组成,但却是复杂的实现技术;也可用复杂的组成,但却是一般速度的实现技术。前者可能要选用高性能的器件,从而增加器件测试、组装、电源和冷却等的负担;而后者可能造成组成设计复杂化和更多地采用专用芯片。组成和实现的权衡取决于器件来源、厂家技术特长和性能价格比能否优化。应当在当时器件技术条件下,保证价格不增或只增很少的情况下尽可能地提高速度。

结构、组成和实现所包含的具体内容随不同时期及不同的计算机系统也会有差异。在某些系统中作为结构的内容,在另一些系统中可能是组成和实现的内容。软件的硬化和硬件的软化都反映了这一事实。VLSI 的发展更使结构、组成和实现融于一体,难以分开。

由于计算机组成和计算机实现关系密切,有人将它们合称为计算机实现,即计算机系统的逻辑实现和物理实现。

结论:计算机系统结构设计的任务是进行软、硬件的功能分配,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,实际包括了系统结构和组成两个方面的内容。因此,它研究的是软、硬件的功能分配以及如何更好、更合理地实现分配给硬件的功能。可把着眼于软、硬件功能分配和确定程序设计者所看到的机器级界面的计算机系统结构,称为从程序设计者看到的计算机系统结构;而把着眼于如何更好、更合理地实现分配给硬件的功能的计算机组成,称为从计算机设计者看到的计算机系统结构。

1.3 计算机系统的软硬件取舍、性能评测及定量设计原理

1.3.1 软、硬件取舍的基本原则

软、硬件的功能分配是计算机系统结构的主要任务,而软件和硬件在逻辑功能上又是等效的。从原理上讲,软件的功能可以用硬件或固件完成,硬件的功能也可以用软件模拟完成,只是它们在性能、价格、实现的难易程度上有所不同。

【例 1-6】 编译程序、操作系统等许多用机器语言子程序实现的功能完全可以用组合电路硬件或微程序固件来解释实现。它们的差别只是软件实现的速度慢,编制复杂,编程工作量最大,程序所占的存储空间较多,这些都是不利的;但是,所需硬件少,硬件实现上也就简单容易,硬件的成本低,解题的灵活性和适应性较好,这些都是有利的。乘法运算可经机器专门设计的乘法指令用硬件电路或乘除部件来实现,也可以通过执行一个使用相加、移位、比较、循环等机器指令组成的机器语言子程序来实现。向量、数组运算在向量处理机中是直接使用向量、数组类指令和流水或阵列等向量运算部件硬的方式来实现的,但在标量处理机上也可以通过执行由标量指令组成的循环程序软的方式来完成。浮点数运算可以直接通过设置浮点运算指令用硬件来实现,也可以用两个定点数分别表示浮点数的阶码和尾数,通过程序方法把浮点数阶码和尾数的运算映像变换成两个定点数的运算,用子程序软的方式完成。十进制运算可以通过专门设置十进制运算类指令和专门的十进制运算部件硬的方式完成,或者通过设置BCD数的表示和若干BCD数运算的校正指令来软硬结合地实现,也可以先经10转2有数制转换子程序将十进制数转换成二进制数,再用二进制运算类指令运算,所得结构又调用2转10的数制转换子程序转换成十进制数这种全软的方式实现。

具有相同功能的计算机系统,其软、硬件功能分配比例可以在很宽的范围内变化,如图1-4所示。这种分配比例随不同时期及同一时期的不同机器动态地改变。由于软、硬件紧密相关,软、硬件界面常常是模糊不清的。例如,很难分清中断处理、存储管理等功能中哪些是硬件完成的,哪些是软件完成的。在满足应用的前提下,软、硬件功能分配的比例主要看能否充分利用硬、器件技术的进展,使系统有高的性能价格比。因此,采用何种方式实现,应从系统的应用、效率、速度、造价、资源状况等多个方面综合考虑,对软件、硬件、固件取舍进行综合平衡。

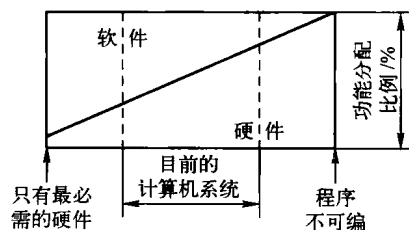


图 1-4 计算机系统的软、硬件功能分配比例

一般来说,提高硬件功能的比例可提高解题速度,减少程序所需存储空间,但会增加硬件成本,降低硬件利用率和计算机系统的灵活性及适应性;而提高软件功能的比例可降低硬件成本,提高系统的灵活性、适应性,但解题速度下降,软件设计费用和所需存储器用量要增加。

原则 1 应考虑在现有硬、器件(主要是逻辑器件和存储器件)条件下,系统要有高的性能价格比,主要从实现费用、速度和其他性能要求来综合考虑。

以下仅从实现费用要求讨论。

无论是硬件实现,还是软件实现,实现费用都应包括研制费用和重复生产费用。目前尽管

软件的设计效率低,而用硬件实现的设计费用还是明显地高于用软件实现的费用,尤其是 VLSI 专用芯片的设计费用是比较高的。

假设某功能的软、硬件实现的每次设计费用分别为 D_s 和 D_h , 则 $D_h \approx 100D_s$ 是完全可能的。

至于重复生产费用,硬件实现的也比软件实现的贵得多,后者只是软件的复制费用加上存放该软件的存储介质(如盘片)的价格。假设该功能软、硬件实现的每次重复生产费用分别为 M_s 和 M_h , 则 $M_h \approx 100M_s$ 也是可能的。

用硬件实现的功能(如子程序调用的全部操作)一般只需设计一次,而用软件实现时,每用到该功能往往要重新设计。设 C 为该功能在软件实现时需重新设计的次数,则该功能用软件实现的设计费用就为 $C \times D_s$ (由于重新设计时可利用原设计进行修改或简单套用,使设计费用 D_s 要低得多)。同一功能的软件在存储介质上可能多次复制和存储。如出现了 R 次,则软件实现此功能的重复生产费用为 $R \times M_s$ 。

假定某计算机系统生产了 V 台。每台计算机用硬件实现的费用为 $D_h/V + M_h$, 若改用软件实现则为 $C \times D_s/V + R \times M_s$ 。只有当

$$D_h/V + M_h < C \times D_s/V + R \times M_s$$

时,用硬件实现才是适宜的。将上述 D_h 与 D_s 、 M_h 与 M_s 的比值代入,得

$$100D_s/V + 100M_s < C \times D_s/V + R \times M_s$$

结论: 只有在 C 和 R 的值较大时,这个不等式才越能够成立。就是说,只有这个功能是经常要用的基本单元功能,才宜于用硬件实现,不要盲目地认为硬件实现的功能比例越大越好。

目前,就软件设计费用来说要远比软件的重复生产费用高, $D_s \approx 10^4 \times M_s$ 也是完全可能的。将此关系式代入上式,得

$$10^6/V + 100 < 10^4 \times C/V + R$$

由于 C 值一般总比 100 小,因此 V 值越大,这个不等式才越成立。

结论: 只有对产量大的计算机系统,增大硬件功能实现的比例才是适宜的。如果用硬件实现不能给用户带来明显的好处,产量仍较低,则系统是不会有生命力的。

原则 2 要考虑到准备采用和可能采用的组成技术,使之尽可能不要过多或不合理地限制各种组成、实现技术的采用。这一点已在 1.2 节中讲过。

原理 3 不能仅从“硬”的角度考虑如何便于应用组成技术的成果和便于发挥器件技术的进展,还应从“软”的角度把如何为编译和操作系统的实现以及为高级语言程序的设计提供更多更好的硬件支持放在首位。

结论: 应当进一步缩短高级语言与机器语言,操作系统与计算机系统结构,程序设计环境(如模块化、数据类型抽象)等与计算机系统结构之间存在的语义差距。计算机系统结构、机器语言是用硬件和固件实现的,而这些语义差距是用软件来填补的。语义差距的大小实质上取决于软、硬件功能的分配,差距缩小了,系统结构对软件设计的支持就加强了。我们将在第 2 章结合数据表示、寻址方式与指令系统的设计和改进来讨论在维持一定语义差距的前提下缩小语义差距的某些途径。

1.3.2 计算机系统的性能评测及定量设计原理

1. 计算机系统的性能评测

多数情况在设计通用计算机系统时,进行软、硬件功能分配总是考虑在满足系统性能前提