

普通高校本科计算机专业特色教材精选·计算机硬件

计算机体系结构

吴艳霞 编著

李静梅 主审

清华大学出版社

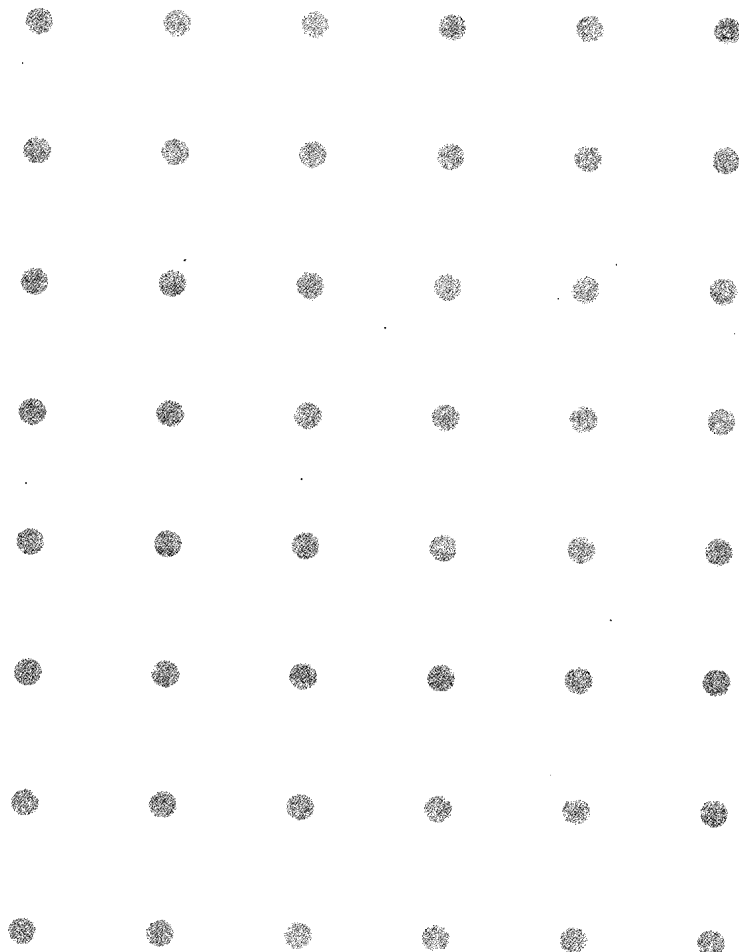


普通高校本科计算机专业特色教材精选·计算机硬件

计算机体系结构

吴艳霞 编著

李静梅 主审



清华大学出版社
北京

内 容 简 介

本书在介绍计算机系统结构的基本概念、原理、结构和分析方法的基础上,着重阐述了计算机系统的并行化技术,旨在帮助学生在建立计算机系统的完整概念,充分掌握计算机系统结构的最新研发思想与技术。

本书共分5章。第1章论述计算机系统的概念及其设计原则,通过对冯·诺依曼计算机模型指令集的分类,简要阐述计算机系统发展的脉络及与体系结构发展密不可分的核心领域的现状与发展,最后简单介绍几种先进的微体系结构;第2章论述流水线的基本概念、分类及性能计算方法,并以DLX模型为实例详细描述流水线执行的关键技术,并对流水线中相关和冲突问题进行重点阐述;第3章通过指令级并行概念的介绍,详细阐述实现指令级并行的关键技术和算法;第4章为“存储系统”,重点讲述虚拟存储系统,Cache存储系统及并行主存系统;第5章详细介绍输入输出系统,讲述总线类型、控制及通信方式,并详细阐述中断系统及总线与CPU及存储器的通信原理。

本书可作为高等院校计算机专业的高年级本科生或研究生的教材,也可作为从事计算机体系结构或嵌入式系统设计的工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机体系结构/吴艳霞编著. —北京:清华大学出版社,2010.10

(普通高校本科计算机专业特色教材精选·计算机硬件)

ISBN 978-7-302-23246-9

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

中国版本图书馆CIP数据核字(2010)第145033号

责任编辑:袁勤勇 张为民

责任校对:梁毅

责任印制:杨艳

出版发行:清华大学出版社

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

社总机:010-62770175

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质量反馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

地址:北京清华大学学研大厦A座

邮编:100084

邮购:010-62786544

印装者:北京鑫海金澳胶印有限公司

经销:全国新华书店

开本:185×260 印张:10.25 字数:233千字

版次:2010年10月第1版 印次:2010年10月第1次印刷

印数:1~3000

定价:19.00元

产品编号:038474-01

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地说，培养具有一定专业特色的和特定能力强的计算机专业应用型人才是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事情。

清华大学出版社在经过大量调查研究的基础上，决定组织出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系，本套教材力求突出学科的理论与实践紧密结合的特征，结合实例讲解理论，使理论来源于实践，又进一步指导实践。学生通过实践深化对理论的理解，更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性，并做到通俗易懂，易教易学，使学生不仅知其然，知其所以然，还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容，学生通过学习和练习，能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力，也注重培养学生解决问题的能力，以适应新经济时代对人才的需要，满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示，学生上机实验指导等辅助教学资源，有些教材还提供能用于网上下载的文件，以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同，所以对特色教学的理解也不尽一致，我们恳切希望大家在使用教材的过程中，及时地给我们提出批评和改进意见，以便我们做好教材的修订改版工作，使其日趋完善。

我们相信经过大家的共同努力，这套教材一定能成为特色鲜明、质量上乘的优秀教材。同时，我们也希望通过本套教材的编写出版，为“高等学校教学质量和教学改革工程”做出贡献。

清华大学出版社

前言

PREFACE

《计算机体系结构》主要作为高等学校计算机专业本科生计算机系统结构课程的教材,在编写此书时力求做到深入浅出,通俗易懂。

现阶段计算机硬件技术以及计算机软件技术互相制约、互相影响促进了两种技术的良性发展,从而使计算机系统结构呈现了迅速发展的态势。本书主要涉及计算机体系结构的概念、结构、机制以及发展,既介绍计算机系统结构方面已经成熟的技术,也介绍国内外最新的研究成果。

作者从五个出发点来编写此书:一、保证书中内容的先进性。在介绍计算机系统设计所必须掌握的知识基础上,以提高效率和系统优化为目标,介绍了现阶段计算机体系结构方面先进的技术方法,如指令级并行技术、同时多线程技术等。二、实例详细,浅显易懂。通过各种代码、实例等具体示例的详细分析和说明,有助于读者阅读。三、合理的知识结构,循序渐进地介绍计算机系统结构的相关知识,同时注重系统性,使读者能够建立起整机的概念。四、结合教学中的重点、难点,加强概念的可理解性,注重学生的知识储备及能力的培养。五、清晰地介绍技术发展的脉络。在介绍原理或算法时,介绍该技术的开发背景、应用情况以及后续发展,开拓读者实践和创新的思维方式,全面展现现阶段计算机体系结构的原理和本质,并紧密结合行业发展,从而促使计算机系统结构课程内容向实用化和新理念方向延伸。

本书共分5章。第1章论述计算机系统的概念及其设计原则,通过对冯·诺依曼计算机模型指令集的分类,简要阐述计算机系统发展的脉络及与体系结构发展密不可分的核心领域的现状与发展,最后简单介绍几种先进的微体系结构;第2章论述流水线的基本概念、分类及性能计算方法,并以DLX模型为实例详细描述了流水线执行的关键技术,并对流水线中相关和冲突问题进行重点阐述;第3章通过指令级并行概念的介绍,详细阐述实现指令级并行的关键技术和算法;第4章为“存储系统”,重

点讲述虚拟存储系统，Cache 存储系统及并行主存系统；第 5 章详细介绍输入输出系统，讲述总线类型、控制及通信方式，并详细阐述中断系统及总线与 CPU/存储器的通信原理。

作者对 Intel 公司在哈尔滨工程大学计算机体系结构精品课程建设方面的大力支持表示衷心的感谢，由于水平有限，书中难免有错误和不妥之处，恳请广大读者批评指正。

作者
2010 年 6 月

目 录

CONTENTS

第 1 章 概述	1
1.1 计算机系统结构的基本概念	1
1.1.1 多级层次结构	1
1.1.2 系统结构、组成和实现之间的关系	3
1.2 计算机系统结构的设计方法	6
1.2.1 计算机系统的设计原则	6
1.2.2 计算机系统的设计思路及步骤	9
1.3 软件、应用、器件的影响	11
1.4 计算机系统结构的分类	15
1.5 基于冯·诺依曼计算机模型的指令集分类	18
1.5.1 CISC 体系结构	19
1.5.2 RISC 体系结构	20
1.5.3 CISC 和 RISC 混合体系结构	22
1.5.4 EPIC 体系结构	22
第 2 章 流水线技术	25
2.1 流水线的概念	25
2.1.1 什么是流水线	25
2.1.2 流水线的分类	26
2.2 流水线的性能指标	29
2.2.1 吞吐率	30
2.2.2 加速比	31
2.2.3 效率	32
2.3 DLX 的基本流水线	33
2.3.1 DLX 指令集结构	33
2.3.2 基本的 DLX 流水线	35
2.3.3 DLX 流水线各级的操作	37
2.3.4 DLX 流水线处理机的控制	39

2.4	流水线的相关与冲突	47
2.4.1	流水线相关	47
2.4.2	流水线冲突	49
第3章	指令级并行	59
3.1	指令级并行的概念	59
3.2	循环展开	60
3.2.1	循环展开的原理	60
3.2.2	循环展开的特点	62
3.3	动态指令调度	63
3.3.1	静态指令调度与动态指令调度	63
3.3.2	动态指令调度的基本思想	63
3.3.3	动态指令调度算法：记分牌	64
3.3.4	动态指令调度算法：Tomasulo 算法	76
3.4	动态分支预测	91
3.4.1	采用分支预测表	92
3.4.2	采用分支目标缓冲器	94
3.4.3	基于硬件的推断执行	96
3.4.4	先进的分支预测技术	102
第4章	存储体系	105
4.1	存储系统的分析	105
4.2	虚拟存储系统	107
4.2.1	地址的映像与变换	107
4.2.2	页面替换算法	112
4.3	Cache 存储系统	114
4.3.1	地址映像与变换方法	115
4.3.2	Cache 替换算法	119
4.3.3	Cache 系统的加速比	122
4.4	并行主存系统	123
4.4.1	单体多字存储器	123
4.4.2	多体交叉存储器	124
第5章	输入输出系统	133
5.1	输入输出原理	133
5.2	总线设计	134
5.2.1	总线的类型	134
5.2.2	总线的控制方式	135
5.2.3	总线通信	137
5.3	中断系统	139
5.3.1	中断方式	139

5.3.2	中断的分类与分级	140
5.3.3	影响中断响应时间的因素	141
5.3.4	中断屏蔽	141
5.4	通道工作原理	143
5.4.1	通道的作用和功能	143
5.4.2	通道种类	144
5.4.3	通道的工作过程	145
5.4.4	通道的流量分析	146
参考文献		149

第 1 章

概 述

CHAPTER

根据摩尔定律,集成电路芯片上所集成的电路的数目,每隔 18 个月就翻一番。如何更合理利用新器件,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,单纯依靠器件的变革是不能解决的,还要靠计算机系统结构上的不断改进。计算机体系结构的发展不断改变我们对计算机的整体认识。由于计算机系统可以看成是按功能划分的多级层次结构,本章首先给出计算机系统结构的定义,并说明结构、组成、实现三者的含义和关系,同时,从计算机系统多级层次结构的角度帮助我们更好地认识计算机系统设计的基本原则与目标;接着通过对冯·诺依曼计算机模型指令集的分类,简要阐述计算机系统发展的脉络;最后简单介绍几种先进的微体系结构。

1.1 计算机系统结构的基本概念

计算机系统由紧密相关的硬件和软件组成,应该怎样从整体上来认识和分析它呢?一种观点是从使用语言的角度将计算机系统看成是按功能划分的多级层次结构。

1.1.1 多级层次结构

随着计算机的发展,计算机语言经历了机器语言(二进制机器指令系统)、汇编语言、高级语言、应用语言这样一个从低级向高级的发展过程,这 4 种语言的后者均以前者为基础,又比前者的功能更强,使用更方便。从这个意义上讲,计算机语言可以分成若干层或若干级,最低层的语言功能最简单。对使用某一层语言编程的程序员来说,只要遵守该级语言的规定,所编写出的程序总是可以在机器上运行并获得结果,而不必考虑程序在机器中究竟是怎样执行的,就好像有了一台直接使用这种语言作为其机器语言的计算机一样。

而实际上,只有二进制机器指令是与机器硬件直接对应,并被其直接识

别和执行的。然而使用机器语言既不方便也无法适应解题需要,更不利于计算机应用范围的扩大。汇编语言是一种符号式程序语言,给程序员编程提供了方便,尽管其每个语句基本上与机器指令对应,却不能被机器硬件直接识别和执行。

那么为什么汇编语言程序可以在机器上运行并获得结果,就好像对汇编语言程序员来说有了一台用汇编语言作为其机器语言的机器呢?可以把这想象成在使用二进制机器语言的实际机器级之上出现了用汇编语言作为机器语言的一级“虚拟”机器,如图 1-1 所示,这样从功能上计算机系统就被看成是一个由虚拟机器 M2 和实际机器 M1 构成的二级层次结构。汇编语言程序员为了能正确编程,只需要熟悉面向他的虚拟机器 M2 即可,不用了解实际机器 M1。在计算机系统上运行汇编语言(L2)源程序,应先将源程序完整地经汇编程序变换成等效的机器语言(L1)目标程序,而后再在实际机器上执行目标程序以获得结果。

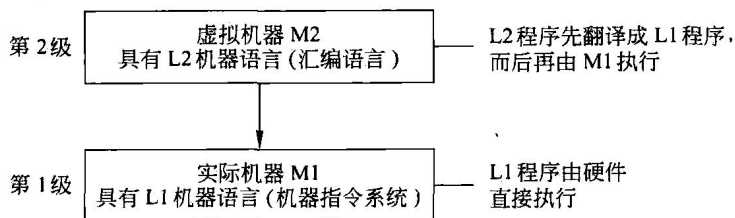


图 1-1 汇编虚拟机的实现

在这种变换模式的递推下,出现了面向题目和过程的高级语言。

这种层次概念还可以引申到机器内部,对于采用微程序控制的机器,每条机器指令都对应于一串微指令(一段微程序),每条微指令执行一些诸如按照各种数据传送通路之类的最基本操作。这样又可以把实际机器级分解成传统机器级和微程序机器级的二级层次结构。微程序机器级的机器语言是微指令系统,当执行到某条机器指令时,控制机器转入执行相应的一串微指令,实现完这条机器指令后,再由程序内的下一条机器指令控制转入实现它的下一串微指令。

翻译和解释是语言实现的两种基本技术。解释比翻译费时,但能节省存储空间。对于微程序控制的机器,在高级语言的实现过程中,先把高级语言源程序经编译程序翻译成传统的机器语言程序,而后再经微程序对每条机器指令进行解释来实现。

那么操作系统应处在这个层次结构中的什么位置呢?从实质上来看,操作系统是传统机器的引申,它要提供传统机器所没有但为汇编语言和高级语言的使用和实现所需的某些基本操作和数据结构,它在许多机器上是经机器语言程序解释实现的。另外,在高级语言机器级之上还可以有应用语言虚拟机器,这种虚拟机所用的语言是面向某种应用环境的应用语言。综上所述,一个现代的计算机系统可以从功能上看成是如图 1-2 所示形式的多级层次结构。

虚拟机器不一定全部由软件实现,有些操作也可以用固件或硬件实现,如操作系统中的某些命令可以由比它低两级的微程序解释,甚至可以设想直接用微程序或硬件来实现高级语言机器。究竟采用何种实现方式,要从整个计算机系统的效率、速度、造价、资源的



图 1-2 计算机系统的多级层次结构

使用状况等方面进行全面考虑,而且需要软件、硬件(包括固件)的综合平衡。实际上,软件和硬件在逻辑功能上是等效的。具有相同功能的计算机系统,其软硬件的功能分配可以在很宽的范围内变化。这种分配比例是随不同时期及同一时期的不同机器而动态变化的。由于软硬件紧密相关,有时软硬件界面也是很模糊的,一个功能很难说哪些是由硬件完成,哪些是由软件完成的。

例如,要在计算机中实现十进制乘法这一功能,既可以用硬件来实现,也可以用软件来完成。

硬件实现方法:设计十进制乘法机器指令,用硬件电路来实现该指令,其特点是完成这一功能的速度快,但需要更多的硬件。

软件实现方法:采用加法、移位等指令通过编程来实现,其特点是实现的速度慢,但不需要增加硬件。

最后,从概念上和功能上把一个复杂的计算机系统看成是由多级构成的层次结构具有很多优点:

- (1) 有利于正确理解软件、硬件、固件在计算机中的地位 and 作用,也有助于理解各种语言的实质和实现途径。
- (2) 直接或间接地推动了计算机系统结构的发展,发展了所谓的高级语言机器。
- (3) 发展了多处理机系统、分布处理系统、嵌入式系统和计算机网络等系统结构。

1.1.2 系统结构、组成和实现之间的关系

1. 计算机系统结构

计算机系统结构也称为计算机体系结构,这个名词从 20 世纪 70 年代开始被广泛采

用,但由于器件技术的发现及计算机软硬件界面的动态变化,使得对它的概念的理解不尽一致。

那么计算机系统结构到底指的是什么?是一台计算机的外表,还是指一台计算机内部的一块块板卡安放结构?1964年,G. M. Amdahl在介绍IBM 360系统时提出:计算机系统结构就是计算机的机器语言程序员或编译程序编写者所看到的外特性,即程序员编写出能在机器上正确运行的程序时必须了解到的概念性结构和功能特性。所谓外特性,就是计算机的概念性结构和功能特性。打一个比方,动物的“系统结构”是指什么呢?它的概念性结构和功能特性就相当于动物的器官组成及其功能特性,如鸡有胃,胃可以消化食物。至于鸡的胃是什么形状的、鸡的胃部是由什么组成的就不是“系统结构”研究的问题了,系统结构只考虑到这一层。

计算机系统结构的外特性一般应包括以下几个方面:

- (1) 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。
- (2) 数据表示:包括硬件能直接识别、处理的数据类型和格式等。
- (3) 操作数的寻址方式:包括最小寻址单位,寻址方式的种类、表示和地址计算等。
- (4) 寄存器的构成定义: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。
- (5) 中断机构和例外条件: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。
- (6) 存储体系和管理:包括最小编址单位、编址方式、主存容量、最大可编址空间等。
- (7) I/O结构:包括I/O的连接方式,设备的访问方式,数据的“源”、“目的”及数据传送量,操作的结束与出错指示等。
- (8) 机器工作状态定义和切换:如管态、目态等的定义和切换。
- (9) 信息保护:包括保护方式、硬件对信息保护的支持等。

关于计算机系统的多层次结构,可以用人与计算机系统作对比,这种对比不是很恰当,但便于更直观地了解计算机系统的多层次结构,如表1-1所示。

表 1-1 计算机系统结构的直观了解

计算机系统	人	计算机系统	人
应用语言级	为人民服务级	传统机器级	人体器官级
高级语言级	读书、学习级	微程序机器级	细胞组织级
汇编语言级	语言、思维级	电子线路级	分子级
操作系统级	生理功能级		

2. 计算机组成与实现

在对计算机组成和实现定义之前,要先介绍计算机系统结构的内部特性,计算机系统结构的内特性就是将那些外特性加以“逻辑实现”的基本属性。这里说的“逻辑实现”就是在逻辑上如何实现这种功能,比如“设计者”给鸡设计了一个一定大小的胃,这个胃的功能是消化食物,这就是鸡系统的某一外特性,那么怎么消化呢?就要通过鸡吃进食物和砂石,再通过胃的蠕动、依靠砂石的研磨来消化食物,这里的吃和蠕动等操作就是内特性。

计算机实现也就是指计算机组成的物理实现。它主要着眼于器件技术和微组装技术。拿上面的例子来说,这个胃由哪些组织组成?由几条肌肉和神经来促使它运动就是相应的“鸡实现”。

据此,我们可以分清计算机系统的外特性、内特性以及物理实现之间的关系。在所有系统结构的特性中,指令系统的外特性是最关键的。因此,计算机系统结构有时就简称为指令集系统结构。

下面介绍对计算机组成和实现的定义。

计算机组成是计算机系统结构的逻辑实现,包括机器级内数据流的组成以及逻辑设计等。它着眼于机器级内各事件的排序方式与控制机构、各部件的功能以及各部件间的联系。

计算机实现是指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。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。

计算机组成设计一般要求确定以下几个方面的内容:

(1) 数据通路宽度:数据总线上一次并行传送的信息位数。

(2) 专用部件的设置:设置哪些专用部件,如乘法专用部件、浮点运算部件、字符处理部件、地址运算部件、每种专用部件的个数等,这些与计算机要求达到的速度、专用部件的使用频率以及成本等因素相关。

(3) 各种操作对部件的共享程度:如果共享程度过高,即使这些操作在逻辑上互不相关,也只能分时使用,这样就限制了速度,可以设置多个部件来降低共享程度,利用提高操作并行度来提高速度,但成本也将由此提高。

(4) 功能部件的并行度:功能部件的控制和处理方式是采用串行,还是采用重叠、流水或分布处理?

(5) 控制机构的组成方式:事件、操作的排序机构是采用硬联控制还是用微程序控制?是采用单机处理还是用多机处理或功能分布处理?

(6) 缓冲和排队技术:在不同部件之间怎样设置及设置多大容量的缓冲器来弥补它们的速度差异?采用什么次序来安排等待处理事件的先后次序,可以是随机、先进先出、后进先出、优先级、循环队列等多种方式?

(7) 预估、预判技术:采用什么原则来预测未来的行为,从而达到优化性能和优化处理的目的?

(8) 可靠性技术:采用什么样的冗余技术和容错技术来提高可靠性?

下面将进一步说明计算机系统结构、计算机组成和计算机实现的概念。

例如,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、取操作数、运算、输送结果等的具体操作及其排序方式属于计算机组成,而实现这些指令的具体电路、器件的设计及装配技术等则属于计算机实现。

又如,确定是否要有乘法指令属计算机系统结构,而乘法指令是用专门的乘法器实现还是经加法器用重复的相加和右移操作来实现则属于计算机组成,而乘法器、加法器的物

理实现,如器件的选定(包括器件集成度、类型、数量等的确定)及采用的微组装技术则属于计算机的实现。

总之,计算机系统结构设计的任务是进行软硬件功能的分配,确定传统机器级的软硬件界面。作为“计算机系统结构”这门学科,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如何分配以及如何最佳、最合理地实现分配给硬件的功能。

1.2 计算机系统结构的设计方法

一般来说,在设计中提高硬件功能的比例可以提高解题速度,减少所需的存储容量,但会提高硬件成本,降低硬件的利用率和计算机系统的灵活性与适应性;而提高软件功能的比例,可以降低硬件的造价,提高系统的灵活性、适应性,但解题的速度会相对下降,软件设计费用和所需的存储器容量需要增加。因此,确定软硬分配比例的一个主要方面应考虑在现有硬件和器件(主要是逻辑和存储器件)的技术条件下,系统要有较高的性价比。

1.2.1 计算机系统的设计原则

1. 局部性原理

程序的局部性原理(principle of locality)是程序最重要的特征。它是指程序总是趋向于使用最近使用过的数据和指令,也就是说程序执行时所访问的存储器地址分布不是随机的,而是相对地簇集,这种簇集包括指令和数据两部分。程序局部性包括程序的时间局部性和程序的空间局部性。

(1) 程序的时间局部性:是指程序即将用到的信息可能就是目前正在使用的信息。若一条指令被执行,则在不久的将来,它可能再被执行。

(2) 程序的空间局部性:是指程序即将用到的信息可能与目前正在使用的信息在空间上相邻或邻近。一旦一个存储单元被访问,那么它附近的单元也可能被很快访问。

程序的局部性原理是计算机体系结构设计的基础之一。利用程序的局部性原理,根据程序最近的访问情况来比较准确地预测将要访问的指令和数据。程序的局部性原理是虚拟存储技术引入的前提。虚拟存储的实现原理是当进程要求运行时,不是将它全部装入内存,而是将其一部分装入内存,另一部分暂不装入内存。

2. 关注经常性事件原则

要关注经常性事件,以经常性事件为重点。在计算机系统设计中,经常需要在多种不同的设计方法之间折中,那么对经常性事件进行优化可以得到更多总体上的改进优化,且效果非常明显,所以在计算机设计中,最重要且应用最广泛的准则就是提高经常性事件的执行速度。在设计上必须有所取舍时,一定要优先考虑经常性事件。

经常出现的事件一般比不经常出现的事件简单,所以提高经常性事件的性能会相对容易些。在计算机设计中应用此原则时,首先要确定哪些是经常性事件,然后分析提高这种情况下的运行速度对计算机的整体性能提高的程度有多少,下面用两个例子来介绍该原则。

例如,处理器中的取指和译码单元要比乘法单元使用更加频繁,那么这两个单元就是

前面所说的“经常性事件”，所以在计算机设计中要首先优化取指和译码单元。如 Pentium M 处理器为了降低系统功耗且同时提高计算机性能，在其译码单元引入“micro-op Fusion”概念，把原有的两个 micro-op (microinstructions) 合成为一个进行操作，提升了传输速度。

又例如，当处理器执行两个数的加法运算时，一般情况下两数相加操作是不会发生溢出的，也就是说不溢出的情况是常见的，溢出情况是不常见的，所以不溢出相加的操作是“经常性事件”，那么在计算机设计中就要对不溢出相加的操作进行优化以达到提高机器性能的目的。当然，对不溢出的相加操作优化后，在处理溢出的相加操作时机器速度就会相对降低，但溢出的情况非常少，总体上机器的性能还是明显提高了。

3. Amdahl 定律

Amdahl 定律的内容是：通过使用某种较快的执行方式所获得的性能提高，受限于该部件占用系统执行时间的百分比。通过改进计算机的某一部分，所得到的性能提升程度可以通过 Amdahl 定律定量地反映出来。该定律将“关注经常性事件原则”进行了量化。

系统性能提升的程度可以用“加速比”这个概念来定量形象地对其进行描述。加速比反映了机器改进前后速度的提升程度，即机器改进前后系统性能提高的程度。如果对某个部件进行了优化改进，那么系统加速比将表达如下：

$$\text{加速比} = \frac{\text{优化后的系统性能}}{\text{优化前的系统性能}} = \frac{\text{优化前的总执行时间}}{\text{优化后的总执行时间}}$$

从加速比的表达式中可以看出加速比主要取决于两个因素：

(1) 优化前的系统中可优化部分的执行时间在总的执行时间中占的比例。例如：一个程序需要运行 60s，其中 20s 的执行部分可以被优化，那么可优化部分所占的比例就是 20/60。我们可将这个值称为“可优化比例”，可优化比例总是小于 1。可优化比例越大，越接近于 1，说明可优化的部分越多，相对得到的加速比就会越大，总体系统性能提高也就会越加明显。

(2) 优化后的系统和优化前的系统性能相比提高的倍数。将可优化部分系统性能提高的倍数称为“优化加速比”。如果一个系统优化后，可优化的部分优化后的执行时间是 2s，而优化前其执行时间是 5s，那么优化加速比就是 5/2。显然，优化加速比是大于 1 的，优化加速比越大，系统性能提高程度越大。

优化前的系统包括可优化部分和不可优化部分，相对地，优化后的系统包括被优化部分和未被优化部分。优化前系统的执行时间等于可优化部分的执行时间和不可优化部分的执行时间之和，优化后系统的执行时间等于未优化部分的执行时间和优化部分的执行时间之和。

$$\begin{aligned} \text{优化后总执行时间} &= t_{\text{未优化部分}} + t_{\text{优化部分}} \\ &= (1 - \text{可优化比例}) \times \text{优化前总执行时间} \\ &\quad + \frac{\text{可优化比例} \times \text{改进前总执行时间}}{\text{优化加速比}} \\ &= \text{优化前总执行时间} \times \left[(1 - \text{可优化比例}) + \frac{\text{可优化比例}}{\text{优化加速比}} \right] \end{aligned}$$

此时系统加速比的表达式如下：