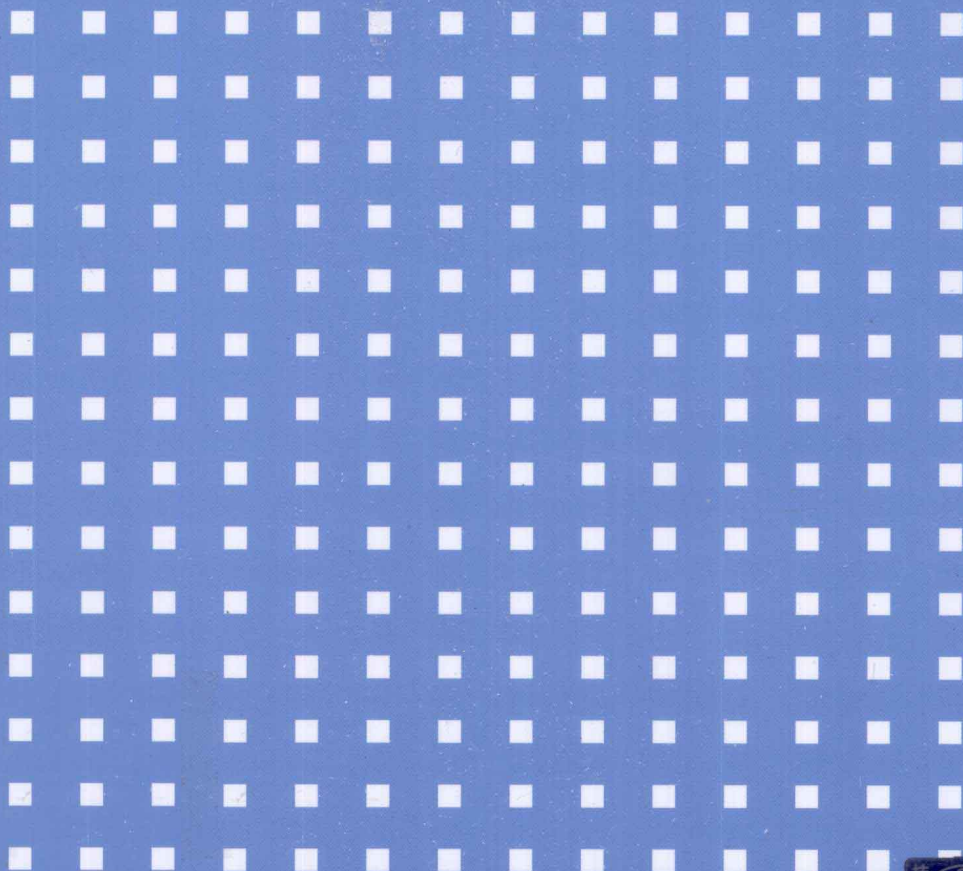


计算机系统结构 (第2版)

李文兵 编著



清华大学出版社

高等学校计算机专业教材精选·计算机原理

计算机系统结构 (第2版)

李文兵 编著

清华大学出版社
北京

内 容 简 介

本教材是计算机专业课程“计算机系统结构”的配套教材。全书围绕如何提高计算机系统性能这一主线,从处理器、存储系统、I/O系统和并行处理系统4个方面,分为15章进行讲解。第1章给出了计算机系统结构的概念和计算机系统性能的定量分析和测试方法。第2章~第6章介绍提高处理器性能的指令系统优化编码方法、流水线技术和向量处理机。第7章、第8章介绍了I/O系统及提高其性能的技术。第9章和第10章介绍了存储系统及提高其性能的各种技术。第11章~第13章介绍了互连函数、互连网络及消息传递机制,分析了消息传递机制的方法和寻径算法。第14章和第15章介绍了多处理器系统和多计算机系统这两个并行处理系统,说明了它们的硬件系统结构和执行软件的设计。

本教材章节篇幅较小,便于教学,适应各种学时安排;本教材所涉及的许多问题,有作者自己的理解或见解;文笔言简意赅,图文共茂,可读性好。

本教材适宜作为计算机或相关专业本科或硕士研究生教材,也可供有关工程技术人员学习参考。

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

计算机系统结构 / 李文兵编著. —2版. —北京:清华大学出版社,2011.10

(高等学校计算机专业教材精选·计算机原理)

ISBN 978-7-302-25843-8

I. ①计… II. ①李… 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(2011)第113528号

责任编辑:汪汉友 薛 阳

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

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

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×260

印 张:13.75

字 数:337千字

版 次:2011年10月第2版

印 次:2011年10月第1次印刷

印 数:1~4000

定 价:23.00元

产品编号:040449-01

出版说明

我国高等学校计算机教育近年来迅猛发展,活学活用计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行了大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材是清华大学出版社从全国各高校的优秀计算机教材中精挑细选的一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下:

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者全部为各校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,相信能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

我们的电子邮件地址是 wanghanyou@tup.tsinghua.edu.cn;联系人:汪汉友。

清华大学出版社

第 2 版前言

《计算机系统结构(第 2 版)》与广大读者见面了。

该版在原版的基础上,增加了 I/O 系统和存储系统两个方面的内容,即第 7 章至第 10 章。为保持原版的模块串接的风格,I/O 系统内容分为第 7 章和第 8 章,存储系统内容分为第 9 章和第 10 章。同时去掉了已经过时的第 1 版中的第 10 章(陈列处理机系统)。这样全书就由 15 章构成。这 15 章系统地介绍了计算机系统结构这门课所要讲授的内容,全面地反映了现代计算机系统的发展成果。

作者对计算机系统结构中的一些问题有自己的理解或见解,且用自己的语言去表述。这些问题包括对计算机系统结构术语和计算机系统结构这门课性质的理解;用图表法构造哈夫曼树的方法以及对 PDP-11 整个指令系统的分析;对流水线概念的表述及其结构和性能的分析;多功能非线性流水线表示及其无冲突调度的步骤、有关术语和状态变换图的理解和表述;互连函数的分析方法以及对互连代数的理解; Ω 互连网络阻塞原因分析及解决方法;对多处理器系统 cache 不一致原因的分析及维护 cache 一致性协议的理解;多计算机系统的构成及其软件方案等。希望这些内容有助于读者理解和掌握相关问题。

本书的章节组织与课堂教学安排相一致,文字言简意赅与教学用语相吻合,书中文图并茂,语言通俗易懂,可读性好,适宜作教学用书。

参加本书修订再版工作的还有王颖、王玉华、李鸿桐、黄硕之、李海迎、贾雯、李海恩、李洪等人。

感谢清华大学出版社对这次修订再版工作所给予的大力支持与帮助。

李文兵
2011 年 5 月

第1版前言

《计算机系统结构》终于与广大读者见面了。

本书由作者讲授“计算机系统结构”课程的教案编写而成,是长期教授这门课的积淀,是学习与研究先进计算机系统的心得,是搞科研项目的体验。希望本书能受到大家的欢迎。

本书的特点是精、顺、透。

“精”是对内容进行了精选。首先,去掉了在前序课“计算机组成原理”中已较为详尽介绍的存储器系统和 I/O 系统两部分内容,其余的内容被归纳为 12 章,力求突出重点,把问题讲清讲透,不求面面俱到。此外,充分利用图与表,文风追求言简意赅。

“顺”是指本书在体现该课程体系和内容的基础上,各章的内容安排、前后衔接力求做到顺畅流畅。这表现在,在内容安排上前为后所用,后用前所有;12 章内容由一条主线贯穿,这就是性能。粗略地说,前 6 章主要介绍时间并行技术,后 6 章主要介绍空间并行,但不管哪项技术都是为了提高计算机系统的性能,每章也都是围绕这条主线展开来讲解的。

“透”就是问题讲得透。作者对这门课所涉及的问题做了较为深入的研究,因此,能把这些问题讲得较为清楚明白,也使得本书有较好的可读性。就在写这个前言时,这些问题仍不断地浮现在作者的脑海里,诸如指令系统的优化设计、多功能非线性流水线的无冲突调度、互连函数与互连代数、互连网络的阻塞、多处理器系统中的 cache 一致性问题。相信读者在读本书时,对这些问题的认识会有所提高,有所收获。

欢迎广大师生将本书选做教材。使用时,建议根据学员的基础与学时计划,每章用 2~4 个学时,总学时数可控制在 30~50 之间。

“计算机系统结构”是一门专业性很强的课程,加之计算机系统及其技术发展又很快,12 章内容只是作者的认识和理解,限于水平,缺点和错误在所难免,欢迎广大读者提出宝贵意见和建议。

参与本书编写工作的还有张景辉、李春华、王玉华、李海迎、黄硕之、贾雯和李海恩等同志。他们对本书的编写都做了一定的工作,在此向他们表示谢意。

作者与清华大学出版社有着长期的良好的合作关系,在出版、发行过程中的各个环节上,包括本书的编写与出版,一直得到清华大学出版社的大力支持与帮助。借此机会向清华大学出版社领导及有关的全体人员表示衷心的感谢。

李文兵
2008 年 5 月

目 录

第 1 章 绪论	1
1.1 计算机系统	1
1.2 计算机系统结构	3
1.3 计算机系统性能的定量分析与测试	5
习题	10
第 2 章 指令系统的优化设计	12
2.1 指令系统	12
2.2 操作码的优化设计	13
2.3 地址码的优化设计	18
2.4 指令系统的优化	21
2.5 复杂指令系统计算机与精简指令系统计算机	24
习题	27
第 3 章 流水线处理机	28
3.1 流水线的概念	28
3.2 流水线结构	31
3.3 线性流水线的性能分析	34
习题	42
第 4 章 流水线的相关处理技术	43
4.1 流水作业的相关问题和冒险	43
4.2 流水线局部相关的处理技术	45
4.3 流水线全局相关的处理技术	48
习题	50
第 5 章 多功能非线性流水线的调度	51
5.1 多功能非线性流水线表示	51
5.2 无冲突调度	52
5.3 流水线调度的优化	54
习题	55
第 6 章 向量处理机	59
6.1 向量及其处理	59
6.2 向量处理机的结构	60
6.3 向量指令	64

6.4 向量处理机的存储器	69
习题	72
第7章 计算机系统总线	73
7.1 系统总线结构	73
7.2 总线标准	75
7.3 总线接口及其标准	83
习题	88
第8章 计算机主机与外设的数据传送方式	89
8.1 程序查询方式	89
8.2 程序中断方式	90
8.3 DMA 方式	95
8.4 通道方式	100
习题	105
第9章 主存与辅存	107
9.1 主存的多体组织	107
9.2 磁盘存储技术	109
9.3 冗余磁盘阵列	113
习题	118
第10章 存储体系	120
10.1 存储系统的层次结构和相联存储器	120
10.2 高速缓冲存储器	123
10.3 虚拟存储器	132
习题	137
第11章 互连函数及互连代数	140
11.1 互连网络的概念	140
11.2 互连函数	142
11.3 互连代数	153
习题	155
第12章 互连网络	156
12.1 网络参数	156
12.2 静态连接网络	157
12.3 动态连接网络	162
习题	171

第 13 章 消息传递机制	173
13.1 消息及其格式.....	173
13.2 消息寻径方式.....	173
13.3 消息寻径算法.....	177
习题.....	183
第 14 章 多处理器系统	184
14.1 系统结构.....	184
14.2 集成方式.....	185
14.3 高速缓存的一致性.....	189
习题.....	194
第 15 章 多计算机系统	196
15.1 集群的优势.....	196
15.2 硬件系统结构.....	196
15.3 并行程序设计.....	199
习题.....	205
参考文献.....	206

第1章 绪 论

作为本书的开篇,本章介绍了计算机系统与计算机系统结构的概念,明确计算机系统结构这门课要学习和研究的问题。

1.1 计算机系统

1. 计算机系统的组成

计算机系统由硬件(hard ware)和软件(soft ware)组成。

(1) 硬件 硬件是指计算机系统在实际装置,包括中央处理器(central processing unit,CPU)、存储器(memory)、外部设备(external devices),以及通道(channel)、总线(bus)等。存储器又分为用来存放正在运行的程序和数据的主存储器(简称主存)和存有大量备用的软件和数据辅助存储器(简称辅存)。中央处理器与主存合称为主机。

(2) 软件 软件是程序及其文档的总称。文档即硬件和程序的有关资料。计算机系统的软件一般认为包括以下几种。

① 系统软件 系统软件是指操作系统、编辑程序、编译系统和诊断程序等软件。

② 计算机语言 计算机语言分为以下3种。

- 高级语言 Fortran、Pascal、Basic 和 Foxpro 等都是高级语言。
- 低级语言 低级语言是指机器语言(指令系统)和汇编语言。
- 中级语言 有人把C语言称为中级语言。

③ 应用程序 应用程序是指用户或软件公司用计算机语言所编写的实际应用程序。

2. 计算机系统的层次结构

计算机系统的层次结构,如图1.1所示,其中裸机就是指计算机系统硬件。

从该层次图应了解到如下两点。

① 指令系统(instruction set)是裸机与软件的接口。计算机硬件(裸机)是根据指令系统设计出来的,因此计算机硬件收到指令系统中的某条指令,就能把这条指令变成实现该指令功能的信号,从而实现该指令的功能。所有软件的功能,都是通过指令系统实现的,也因为如此,人们把用指令的二进制编码所编写的程序称为目标程序。这就是说,所有软件都必须变成目标程序,才能被计算机硬件所识别。

② 操作系统是人机接口。汇编程序、编译程序、编辑程序等都必须在操作系统上才能工作。因此,称操作系统是计算机系统的操作平台。在这个平台上,用户可以使用编辑程序,根据某种语言的语法,编写该语言的应用程序,这样的程序叫源程序。源程序经汇编程序或相应的编译程序翻译(translate),变成对应的目标程序后,再用链接程序把目标程序与相关信息链接在一起,就变成了可执行程序。

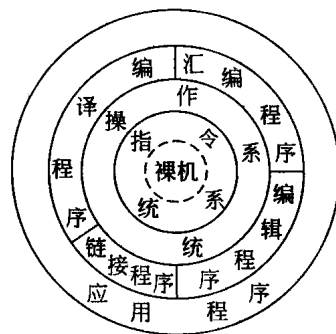


图 1.1 计算机系统的层次结构

3. 计算机系统的分类

关于计算机系统的分类方法,目前使用较多的是弗林(Michael J. Flynn)提出的方法。

1966年,弗林提出指令流(instruction stream)和数据流(data stream)的多倍性(multiplicity)概念,并依此把计算机系统分为4类。

- ① 单指令流单数据流(single instruction stream single data stream, SISD)系统。
- ② 单指令流多数据流(single instruction stream multiple data stream, SIMD)系统。
- ③ 多指令流单数据流(multiple instruction stream single data stream, MISD)系统。
- ④ 多指令流多数据流(multiple instruction stream multiple data stream, MIMD)系统。

多倍性是指在系统性能瓶颈部件上,处于同一执行阶段的指令或数据的最大可能个数。由此而划分的这4类系统,如图1.2所示。图中CU、PU、MU、MM、IS、DS、CS和SM分别为控制部件、处理部件、存储部件、存储模块、指令流、数据流、控制流和共享存储器。

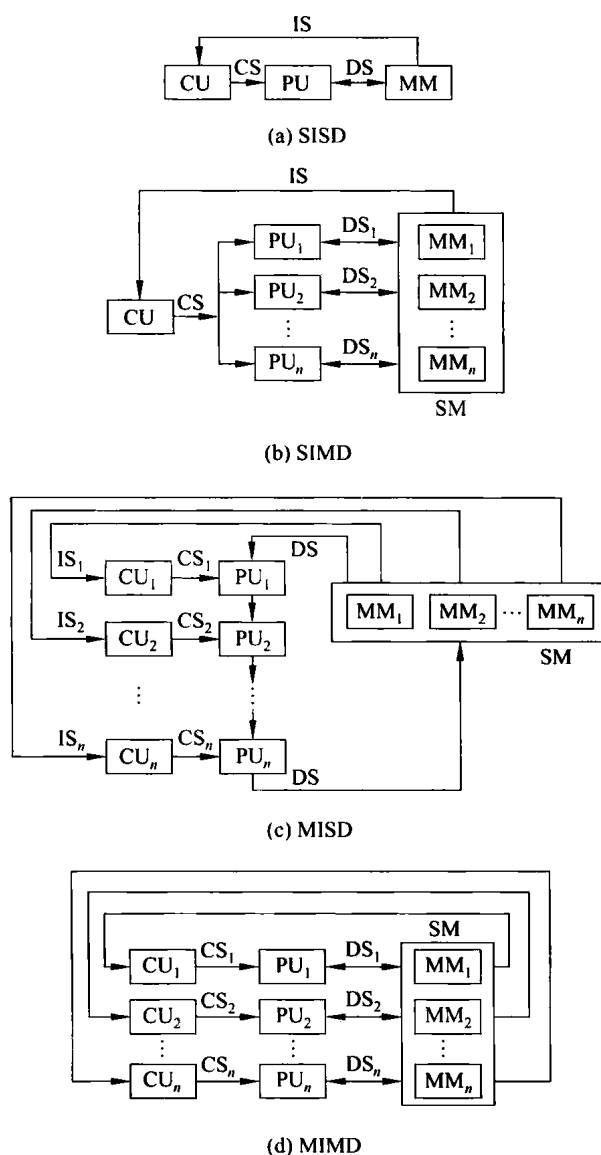


图 1.2 计算机系统的弗林分类法

1.2 计算机系统结构

1. 计算机系统结构(computer architecture)术语

(1) 从计算机系统的设计角度来看,计算机的一般设计过程如下:

- ① 确定计算机的功能、性能和价位;
- ② 设计指令系统;
- ③ 结构设计,包括存储系统、总线结构、I/O 系统,以及内部 CPU 结构等;
- ④ 硬件设计,主要是硬件逻辑设计及其芯片的封装技术。

站在计算机设计者的角度来看,计算机系统结构指的就是第②步要完成的任务,即指令集(instruction set)的系统结构,这是计算机软件与硬件的界面。关于这一看法,在美国斯坦福大学教授 John L. Hennessy 与加州大学伯克利分校教授 David A. Patterson 合著的 *Computer Architecture: a quantitative approach* 一书的第 3 版中写道,计算机系统结构这一术语通常仅指指令集设计。又说,指令系统结构是指实际程序员所见的指令集,这一指令集的系统结构担当着软件与硬件之间的界面。

(2) 从计算机系统的应用角度来看,这里有必要再强调一下,计算机系统结构指的是机器语言级的程序员所看到的计算机属性,即概念性结构与功能特性。

这里,实际上是强调了以下两点。

① 不同级别的程序员所看到的计算机系统将具有不同的属性。例如,高级程序员看 NOVA 机(双总线结构)和 PDP-11 机(单总线结构)几乎没有什么差别,具有相同的属性,而它们的差别是看不出来的。这时,NOVA 机与 PDP-11 机的差别,对于高级程序员来说具有透明性(transparency)。透明性是计算机学科中一个重要概念,是指本来存在的事物或属性,如果从某个角度去看,好像不存在,就称之为透明。同样是 NOVA 与 PDP-11 这两种机型,让机器语言级程序员去看,属性却不同,这主要表现在指令系统及其相关的寄存器结构上。因此计算机系统结构强调的是,机器语言级程序员所看见的计算机属性。应该认为,Hennessy 与 Patterson 在定义计算机系统结构时所说的实际程序员指的就是机器语言级程序员。

② 与从计算机系统设计者的角度相比,这里强调的是功能性结构,是从机器语言级程序员角度所看到的计算机软件与硬件交界面的结构。这实际上还是指指令系统及其相关的功能结构。

总之,无论从哪个角度看,计算机系统结构这一术语是专指计算机系统中软件与硬件交界面的结构及其功能,它所指的这一结构层面属于硬件范畴。这样,对于计算机系统设计者来说,就要研究计算机软件与硬件的功能分配,并确定它们的界面,就是确定哪些功能由软件实现,哪些功能由硬件实现。硬件实现其功能是靠指令驱动的,而软件功能的实现,也要把软件编译为其所对应的指令序列,可见,软件与硬件之间的这个界面只能是指令系统结构,即计算机系统结构。因此可以说,计算机系统结构的设计,是计算机系统设计的关键环节。其设计追求的目标如下:

- 高性能/价格比;
- 大吞吐量;

- 低系统开销；
- 短作业运行时间。

对于机器语言级程序员来说,必须了解指令系统及其相关结构,因此,研究的内容应包括:

- 机器数表示;
- 处理器的寄存器组织;
- 存储系统;
- 机器级 I/O 结构,等等。

【例 1.1】 从机器语言程序员的角度看,下列哪些器件是透明的: 通用寄存器、状态标志寄存器、指令寄存器、时序发生器、主存地址寄存器、移位寄存器、计数器、计时器和加法器。

解 从机器语言级程序员的角度看,所谓透明,就是这些器件与计算机系统结构这个层面无关。显然,这些器件是指令寄存器、时序发生器、主存地址寄存器、移位寄存器、计数器、计时器和加法器。

2. 计算机系统结构学科

计算机系统结构,也称计算机体系结构。在学科领域,“计算机系统结构”是一个学科,属计算机学科中的一个二级学科。

(1) 计算机系统结构学科的知识体系。如图 1.3 所示,图中左边列出了计算机系统结构学科的知识结构,右边是知识所对应的课程设置。在我国,对应计算机系统结构学科,一般设置有 5 门课程。按内容衔接,它们的教学次序是“数字逻辑”、“计算机组成原理”、“汇编语言程序设计”、“接口与通信”、“计算机系统结构”。

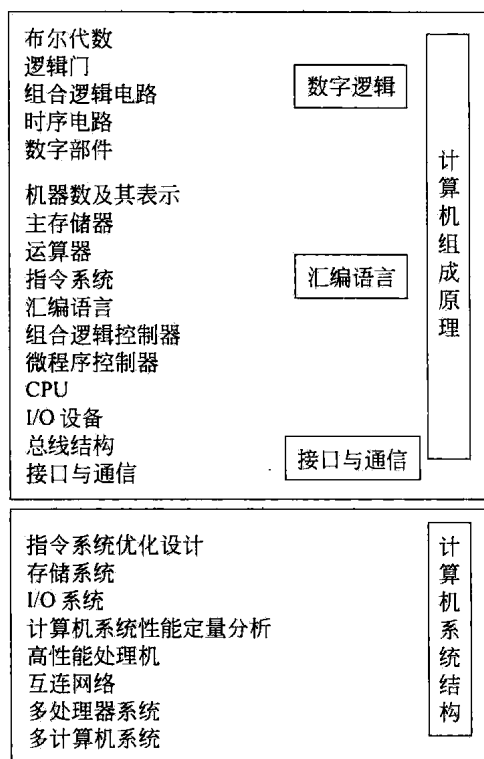


图 1.3 计算机系统结构的知识体系及其课程设置

可以说,这 5 门课的内容,基本上覆盖了计算机系统结构学科的知识结构。

(2) “计算机系统结构”课与“计算机组成原理”课的关系与差别。“计算机系统结构”课是“计算机组成原理”课的教学深入与提高,两门课是前序课与后续课的关系。因此,这两门课所介绍的内容有很大的关联性,甚至出现内容重复。但毕竟是两门课,它们应各有侧重,作为后续课的“计算机系统结构”课(简称后课)应突出在内容深化、技术发展、定量分析,以及系统观点上。这两门课的差别如下:

① “计算机组成原理”课(简称前课)重在功能及原理的分析与设计;而后课侧重性能的评估与优化。因此,后课所讲的内容,多属于高性能计算机,诸如流水线技术、向量处理机、互连网络和多处理器系统。

② 前课讲的内容,一般是定性的;而后课强调定量分析。这样,后课的内容就涉及指令编码的优化技术、CPU、存储系统,以及 I/O 系

统的性能定量计算与分析。

③ 前课主要是介绍计算机主机,及其功能部件的逻辑实现,涉及 I/O 设备及 I/O 方法;而后课着力培养学员的系统观点,比如分析计算机系统的性能,不仅要看 CPU 的性能,还应考虑到包括存储系统、I/O 系统在内的整个计算机系统的性能,这样才能知道计算机系统的真实性能。

本书的内容设置就是根据“计算机系统结构”的学科体系,以及认识规律而确定的,正如全书目录所示。笔者认为,这种安排是科学的;但并不认为是唯一的,因为课程设置及其内容安排是与教学资源及授受对象有很大关系的。实际上,从国外情况看,“计算机系统结构”学科的课程设置及内容安排的版本很多。这是很正常的,也是很好的现象。

1.3 计算机系统性能的定量分析与测试

1. 定量分析方法

计算机性能的衡量标准有运算速度、CPU 时间和执行时间等。这里只介绍这 3 种。

(1) 运算速度。一般来说,计算机性能越好,其运算速度就应越快。因此,人们很自然就用运算速度的快慢来衡量计算机性能的好坏。常见的运算速度的单位有如下两种。

① MIPS (million instruction per second, 百万条指令每秒), 人们把它作为衡量计算机运行快慢的一种标准单位, 即 MIPS 值大的, 就认为快; 而小的, 就认为慢。这样, 比较计算机性能的问题, 就变成了求计算机的 MIPS 值。对于给定的程序来说, 如果知道其所含的指令条数 (instruction count, IC) 和执行时间 (execution time, ET), 那么, 所对应的 MIPS 值为

$$\begin{aligned} \text{MIPS} &= \frac{\text{IC}}{\text{ET} \times 10^6} = \frac{1}{\frac{\text{ET}}{\text{IC}} \times 10^6} \\ &= \frac{1}{\text{平均每条指令的执行时间} \times 10^6} \end{aligned} \quad (1.1)$$

一般计算机都给出其时钟周期 (clock cycle) 或时钟频率 (clock rate), 它们是互为倒数的关系, 即

$$\text{时钟周期} = \frac{1}{\text{时钟频率}} \quad (1.2)$$

如果知道平均每条指令的时钟周期数 (clock cycles per instruction, CPI), 那么, 平均每条指令的执行时间就可以表示为

$$\text{平均每条指令的执行时间} = \text{时钟周期} \times \text{CPI} = \frac{\text{CPI}}{\text{时钟频率}} \quad (1.3)$$

于是, MIPS 值公式也可以记作

$$\text{MIPS} = \frac{\text{时钟频率}}{\text{CPI} \times 10^6} \quad (1.4)$$

式(1.1)和式(1.4)使用起来都比较方便, 其中式(1.4)更为简便。

【例 1.2】 某微处理器, 其主频为 20MHz, 若平均每条指令用 3 个机器周期时间, 每个机器周期由两个时钟周期组成, 请计算该处理器的平均运行速度。

解 依题给,已知

$$\text{时钟频率} = 20\text{MHz}$$

$$\text{CPI} = 2 \times 3$$

故

$$\text{MIPS} = \frac{20 \times 10^6}{2 \times 3 \times 10^6} \approx 3.33$$

即该处理器的平均运行速度为每秒执行 3.33×10^6 条指令。

用 MIPS 表示机器性能的优点是比较符合常理,人们容易理解;缺点如下。

- 同一台计算机,运行不同的程序,一般来说,会有不同的 MIPS 值。
- 不同的计算机,指令系统不同,用 MIPS 值做比较,未必能说明问题。

② MFLOPS (million floatingpoint operations per second, 百万条浮点操作每秒),也是一种常用的衡量计算机性能的标准单位。目前,高性能计算机一般用它来表示性能。MFLOPS 是基于浮点操作个数,而不是指令条数,它的值是由某个程序所含浮点操作的个数除以该程序的运行时间得到的。因此, MFLOPS 标准存在以下的问题。

- 因为执行时间是衡量机器性能的根本标准,从 MFLOPS 值的求法可知,求 MFLOPS 值没有必要。
- 不同的计算机,浮点指令集不同,操作类型也不相同,所以, MFLOPS 值并不能准确地反映机器性能。
- 单个程序的 MFLOPS 值并不能准确地反映一台计算机的性能。不同的程序,浮点操作与整数操作的比例不同,所用到的浮点操作的快慢也会不同,因此,所得出的 MFLOPS 值就会不同。一般厂商提供的 MFLOPS 值,都是用速度快的浮点操作算出来,其可信度应打折扣。

(2) CPU 时间 (CPU time)。人们习惯于用速度来说明性能,其实,时间才是衡量性能最可靠的标准。因此,专家们引进了术语 CPU 时间。所谓 CPU 时间是指 CPU 的工作时间,换句话说,是执行程序占用 CPU 的时间。显然,一个程序的 CPU 时间为

$$\begin{aligned}\text{CPU 时间} &= \text{时钟周期} \times \text{时钟数} \\ &= \text{时钟周期} \times \text{CPI} \times \text{IC} \\ &= \frac{\text{CPI} \times \text{IC}}{\text{时钟频率}}\end{aligned}\quad (1.5)$$

式(1.5)中,CPI 为每条指令的平均时钟周期数。如果知道包含有 n 条指令的程序中每条指令的时钟数和执行次数,其中第 i 条指令的为 CPI_i 和 IC_i ,那么,执行该程序的 CPU 时间可以表示为

$$\text{CPU 时间} = \text{时钟周期} \times \sum_{i=1}^n (\text{CPI}_i \times \text{IC}_i) \quad (1.6)$$

在能得到每条指令在程序执行中出现的频度 (frequency, 也被称为混合比),以及每条指令所需时钟数的实际测量值时,CPI 值可用式(1.7)求解。

$$\text{CPI} = \sum_{i=1}^n (\text{CPI}_i \times \text{频度}) \quad (1.7)$$

CPU 时间可由此算出

$$\text{CPU 时间} = \text{时钟周期} \times \sum_{i=1}^n (\text{CPI}_i \times \text{频度}) \times \text{IC} \quad (1.8)$$

式中, $n=IC$, 即总指令条数。

【例 1.3】 设有两个 CPU, 它们对条件转移指令采取了两种不同的设计方法。CPU_A 采用一条比较指令来设置相应的条件码, 由随后的一条转移指令测试该条件码, 以确定是否转移, 即实现一次条件转移要执行比较、测试转移两条指令。CPU_B 采用比较、测试转移合为一条指令。假定两个 CPU 执行条件转移指令都需要两个时钟周期, 其他指令只需一个时钟周期, CPU_A 的转移指令条数占 20%, 而 CPU_B 的时钟周期比 CPU_A 的要慢 25%, 请比较这两个 CPU 的性能。

解 两个 CPU 的性能可用它们的 CPU 时间来衡量, CPU 时间短的, 其性能就好。

① 比较 IC 由于两个 CPU 的区别, 仅仅是 CPU_B 比 CPU_A 少 20% 的比较指令, 所以

$$IC_B = (1 - 20\%)IC_A = 0.8IC_A$$

② 比较 CPI 因为 CPU_B 的总指令条数 $IC_B = 0.8IC_A$, 其条件转移指令数是 CPU_A 总指令条数 IC_A 的 20%, 所以, CPU_B 条件转移指令占其总指令条数的比例为

$$\frac{20\% \times IC_A}{0.8 \times IC_A} = 25\%$$

显然, CPU_B 的其他指令所占比例为

$$1 - 25\% = 75\%$$

于是

$$CPI_B = 2 \times 25\% + 1 \times 75\% = 1.25$$

而

$$CPI_A = 2 \times 20\% + 1 \times 80\% = 1.20$$

$$\frac{CPI_B}{CPI_A} = \frac{1.25}{1.20}$$

所以

$$CPI_B = \frac{1.25}{1.20}CPI_A$$

③ 比较时钟周期 因为 CPU_B 的时钟周期要比 CPU_A 的慢 25%, 所以

$$\text{时钟周期}_B = (1 + 25\%) \text{时钟周期}_A = 1.25 \text{ 时钟周期}_A$$

④ 比较 CPU 时间 根据步骤①~步骤③计算结果, 可得

$$\begin{aligned} \text{CPU 时间}_B &= \text{时钟周期}_B \times CPI_B \times IC_B \\ &= 1.25 \times \text{时钟周期}_A \times \frac{1.25}{1.20}CPI_A \times 0.8IC_A \\ &= \frac{1.25}{1.20} \times \text{时钟周期}_A \times CPI_A \times 0.8IC_A \\ &= \frac{1.25}{1.20} \times \text{CPU 时间}_A \end{aligned}$$

即

$$\frac{\text{CPU 时间}_B}{\text{CPU 时间}_A} = \frac{1.25}{1.20}$$

根据分析与计算结果, CPU_A 性能较为优越。

CPU 时间根据 CPU 执行的程序不同, 可分为用户 CPU 时间和系统 CPU 时间, 分别是

指 CPU 执行用户程序的时间和执行用户程序所需的操作系统功能调用的时间。CPU 时间应该是两者之和。但由于系统 CPU 时间很难测量或准确计算,所以往往被忽视。如果使用 UNIX 操作系统,用它的时间命令,可得到系统 CPU 时间。

(3) 执行时间(execution time),也被说成经过时间(elapsed time)、响应时间(response time)。由于 CPU 时间有较好的可计算性,而被应用;但实际上,它并不能反映整个计算机系统的性能。因为计算机完成一项任务所需时间,除了 CPU 时间外,还包括以下时间。

- 访存等待时间(也叫 CPU 暂停时间)。它是指 CPU 访问存储器,而存储器不能立即提供数据时,CPU 被迫暂停工作的时间。
- I/O 机构耗时。它是输入或输出时,I/O 机构操作所耽误的时间。
- 操作系统开销。它是指操作系统运行时,除执行用户程序外,CPU 额外所花费的时间。
- 外部总线延迟。CPU 内部总线的延迟时间虽然很短,但是也算在指令执行周期之内;外部总线的延迟就相对较长,有时是个不可忽略的时间。

为此,专家们又引进了执行时间术语。它是指计算机完成一项任务所需的全部时间。它全面地反映了任务执行过程中各个阶段的延迟,真实地反映出整个系统的时间开销,其值等于上述各项时间之和,即

$$\begin{aligned} \text{执行时间} = & \text{CPU 时间} + \text{访存等待时间} + \text{I/O 机构耗时} + \\ & \text{操作系统开销} + \text{外部总线延迟} \end{aligned} \quad (1.9)$$

其中,前两项之和被称做 CPU 执行时间,即

$$\begin{aligned} \text{CPU 执行时间} = & \text{CPU 时间} + \text{访存等待时间} \\ = & \text{CPU 时间} + \text{等待次数} \times \text{等待代价} \end{aligned} \quad (1.10)$$

等待代价就是每次等待所需的时间,代价也可以用时钟数来表示。

应该说,执行时间是衡量机器性能最可靠的标准。性能与执行时间成反比,即

$$\text{性能} = \frac{1}{\text{执行时间}} \quad (1.11)$$

可见,要提高机器性能,就要缩短其执行任务的时间。从执行时间公式可以看出,缩短执行时间的办法就是提高计算机系统各部分的效率。具体如下:

- 提高 CPU 性能。
- 减少访存等待时间,办法就是减少等待次数,或减少每次等待的代价,当然两者同时都减少就更有效。最好是减到没有等待时间,这样的计算机叫零等待计算机,人们一直在向这个方向努力。
- 减少 I/O 机构耗时,为此,人们在从两个方面解决这个问题:一是更新换代 I/O 设备,提高 I/O 设备运行速度,并缩短其辅助操作时间;二是改进 I/O 方式,最有效的方法就是采用 I/O 与 CPU 并行工作方式。
- 采用或开发高效操作系统。
- 提高外部总线传输效率。

当外部总线传输效率较高而不用考虑,I/O 操作又与 CPU 并行工作时,式(1.9)就可以简化为

$$\text{执行时间} = \text{CPU 执行时间} + \text{操作系统开销} \quad (1.12)$$