



普通高等教育“十一五”国家级规划教材

计算机系统结构

(第2版)

张晨曦 王志英 等 编著

高等教育出版社



清华大学出版社 清华大学计算机专业系列教材

计算机系统结构

（第2版）

唐朔飞 编著 李国忠 主审

清华大学出版社



普通高等教育“十一五”国家级规划教材

计算机系统结构

Jisuanji Xitong Jiegou

(第2版)

张晨曦 王志英 张春元 王伟 沈立 编著

高等教育出版社·北京

内容提要

本书系统地论述了计算机系统结构的基本概念、基本原理、基本结构 and 基本方法,并强调采用量化的分析方法,使读者能够更具体、实际地分析和理解计算机系统结构。

全书共 10 章。其中,第 1 章论述计算机系统结构的基本概念以及定量分析基础;第 2 章论述计算机指令集的设计、RISC 技术及 MIPS 指令集结构;第 3 章论述流水线及其性能分析、流水线中的相关和冲突及其解决方法、向量处理机;第 4 章论述指令级并行性开发,包括指令动态调度、指令分支预测、Tomasulo 算法、前瞻执行、超标量、超流水以及超长指令字技术等;第 5 章论述 Cache 的基本知识、降低 Cache 失效率的方法、减少 Cache 失效开销的方法以及减少命中时间的方法;第 6 章论述总线、通道处理机及其流量分析、廉价磁盘冗余阵列 RAID;第 7 章论述互连函数、互连网络的结构参数与性能指标、静态互连网络、动态互连网络及消息传递机制等;第 8 章论述对称式共享存储器系统结构、分布式共享存储器系统结构、多处理机 Cache 一致性、同步及同时多线程技术;第 9 章论述机群的结构、软件模型及分类;第 10 章论述多核系统结构、实例及基于多核的并行程序设计。

本书内容丰富,实例具体,语言简练,可读性好,并配有大量寓教于乐的动画课件,可作为高等院校计算机、软件工程、自动化及电子工程等相关专业本科生和研究生的教材,也可供相关科技人员参考。

图书在版编目(CIP)数据

计算机系统结构/张晨曦等编著. -- 2 版. -- 北京:高等教育出版社,2014. 8

ISBN 978 - 7 - 04 - 040388 - 6

I. ①计… II. ①张… 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(2014)第 136203 号

策划编辑	倪文慧	责任编辑	倪文慧	封面设计	于文燕	版式设计	王艳红
插图绘制	杜晓丹	责任校对	李大鹏	责任印制	朱学忠		

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.landracom.com
印 刷	高教社(天津)印务有限公司		http://www.landracom.com.cn
开 本	787mm × 1092mm 1/16		
印 张	23.5	版 次	2008 年 6 月第 1 版
字 数	480 千字		2014 年 8 月第 2 版
购书热线	010 - 58581118	印 次	2014 年 8 月第 1 次印刷
咨询电话	400 - 810 - 0598	定 价	38.00 元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 40388 - 00

前 言

本书第1版是“十一五”国家级规划教材,2009年被评为普通高等教育精品教材。为了跟踪和反映计算机技术的新发展,我们对本书进行了修订,补充了新内容,在很大程度上进一步提高了可读性和系统性,并开发了相应的多媒体动画课件。

本书可作为高等院校计算机、软件工程、自动化及电子工程等相关专业本科生和研究生的教材。在内容的选择上,本书不打算覆盖计算机系统结构的所有方面,不想成为厚而全的参考手册,而是重点论述现代大多数计算机都采用的比较成熟的思想、结构和方法。

计算机系统结构强调从总体结构、系统分析这一角度来研究计算机系统。通过学习本书,读者能把在“计算机组成原理”、“数据结构”、“操作系统”、“汇编语言程序设计”等课程中所学的软、硬件知识有机地结合起来,从而建立起计算机系统的完整概念。

本书的主要特色有:

- 重点论述现代大多数计算机都采用的比较成熟的思想、结构和方法,重点突出,避免面面俱到。
- 借鉴国际上最经典的系统结构教材,强调采用量化的分析方法,使读者能更具体、实际地分析和设计计算机系统结构。
- 配有基于动画解析教学法的近200个动画课件及PPT讲稿(见配书光盘),使用它们进行教学或者自学能达到非常好的效果。
- 配有基于自主开发的模拟器的实验(见《计算机组成原理与系统结构模拟实验教程》(高等教育出版社,即将出版),对于深入理解和掌握本课程内容有很大的作用。
- 配有《计算机系统结构学习指导与题解》(高等教育出版社出版)。

全书共包括以下10章。

第1章 计算机系统结构的基本概念,论述计算机系统结构的概念、系统结构和并行性的发展,并介绍定量分析基础。

第2章 计算机指令集结构,论述计算机指令集结构设计中的一些问题,包括寻址技术、指令集的功能设计、操作数的类型和大小、指令格式的设计,并介绍RISC技术以及一个指令集结构的实例——MIPS。

第3章 流水线技术,论述流水线的基本概念、分类和性能指标,介绍流水线中的相关和冲

突问题及其解决方法等,并对向量处理机的结构及特点、关键技术及性能评价进行讨论。

第4章 指令级并行,论述利用硬、软件技术开发程序中指令间并行性的技术和方法,包括指令动态调度、指令分支预测、Tomasulo 算法、前瞻执行、循环展开、超标量、超流水以及超长指令字技术等。

第5章 存储系统,论述 Cache 的基本知识、降低 Cache 失效率的方法、减少 Cache 失效开销的方法以及减少命中时间的方法,并对虚拟存储器做简要的讨论。

第6章 输入/输出系统,论述总线及其与 CPU/存储器的连接、通道处理机及其流量分析、廉价磁盘冗余阵列 RAID。

第7章 互连网络,论述互连函数、互连网络的结构参数与性能指标、静态互连网络、动态互连网络及消息传递机制等。

第8章 多处理机,论述对称式共享存储器系统结构、分布式共享存储器系统结构、多 Cache 一致性、同步及同时多线程技术以及大规模并行处理机等,并介绍了两个多处理机实例。

第9章 机群系统,论述机群的结构、软件模型以及机群的分类,并介绍了4个典型的机群系统。

第10章 多核系统结构与编程,论述多核系统结构、实例以及基于多核的并行程序设计。

作为教材,使用者可以根据自己的需求选取相应的内容进行教学。有以下3种建议方案。

第一方案:全部10章的内容。根据具体要求,可以安排60小时或更多的教学课时。

第二方案:第1章 → 第2章 → 第3章 → 第5章 → 第6章 → 第7章 → 第8章
→ 第10章

这个方案去掉了难度较大的第4章和第9章,安排48~54小时的教学课时为宜。

第三方案:第1章 → 第2章 → 第3章 → 第5章 → 第7章 → 第8章 → 第10章

对于仅仅需要建立基本系统结构概念的读者,可以采用本方案。本方案在对第3章、第5章和第7章内容略作删减的情况下,只需约36教学课时。如果读者已经了解指令集的有关概念,还可以跳过第2章而安排约32学时的短课程。

本书大部分内容由同济大学张晨曦编写,包括第1~3章、第5~7章的全部内容以及第4章的部分内容。国防科学技术大学王志英编写了第8章,张春元编写了第4章的部分内容,沈立编写了第9章。同济大学王伟编写了第10章。

本书直接或间接地引用了许多专家和学者的文献或著作,在此表示衷心的感谢。

本书的教学课件由张晨曦设计和开发。课程网站 www.arch365.net 上将不断发布与更新有关的教学资源,敬请关注。



“系统结构”微信公众号的二维码

本书的教学资源也可从“系统结构”微信公众号(arch365)获得。请扫描相应二维码,关注后按提示操作。

由于作者水平有限,书中难免有错误和不妥之处,敬请读者批评指正。

欢迎访问张晨曦的个人摄影网站 www.FotoSky.com。“最美景”微信公众号:ZuiMeiJing。



“最美景”微信公众号的二维码

作者

2014年5月于上海

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》,其行为人将承担相应的民事责任和行政责任;构成犯罪的,将被依法追究刑事责任。为了维护市场秩序,保护读者的合法权益,避免读者误用盗版书造成不良后果,我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为,希望及时举报,本社将奖励举报有功人员。

反盗版举报电话 (010)58581897 58582371 58581879

反盗版举报传真 (010)82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第 1 章 计算机系统结构的基本概念	1
1.1 引言	1
1.2 计算机系统结构的概念	2
1.2.1 计算机系统的层次结构	2
1.2.2 计算机系统结构的定义	3
1.2.3 计算机组成和计算机实现	4
1.2.4 计算机系统结构的分类	5
1.3 定量分析技术	7
1.3.1 计算机系统设计的定量原理	7
1.3.2 计算机系统的性能评测	11
1.4 计算机系统结构的发展	15
1.4.1 冯·诺依曼结构	15
1.4.2 软件对系统结构的影响	18
1.4.3 器件发展对系统结构的影响	19
1.4.4 应用对系统结构的影响	20
1.5 计算机系统结构中并行性的发展	21
1.5.1 并行性的概念	21
1.5.2 提高并行性的技术途径	22
1.5.3 单机系统中并行性的发展	23
1.5.4 多机系统中并行性的发展	23
习题 1	24
第 2 章 计算机指令集结构	26
2.1 指令集结构的分类	26
2.2 寻址方式	29
2.3 指令集结构的功能设计	32
2.3.1 CISC 指令集结构的功能设计	34

2.3.2 RISC 指令集结构的功能设计	37
2.3.3 控制指令	39
2.4 操作数的类型和大小	40
2.5 指令格式的设计	42
2.6 MIPS 指令集结构	43
2.6.1 MIPS 的寄存器	43
2.6.2 MIPS 的数据表示	44
2.6.3 MIPS 的数据寻址方式	44
2.6.4 MIPS 的指令格式	44
2.6.5 MIPS 的操作	46
2.6.6 MIPS 的控制指令	47
2.6.7 MIPS 的浮点操作	48
习题 2	49
第 3 章 流水线技术	50
3.1 流水线的基本概念	50
3.1.1 什么是流水线	50
3.1.2 流水线的分类	52
3.2 流水线的性能指标	56
3.2.1 吞吐率	56
3.2.2 加速比	59
3.2.3 效率	59
3.2.4 流水线性能分析举例	60
3.2.5 流水线设计中的若干问题	62
3.3 流水线的相关与冲突	63
3.3.1 一个经典的 5 段流水线	63
3.3.2 相关与流水线冲突	66
3.4 流水线的实现	82

3.4.1 MIPS 的一种简单实现	82	5.2.3 查找方法	161
3.4.2 基本的 MIPS 流水线	85	5.2.4 Cache 的工作过程	163
3.5 向量处理机	90	5.2.5 替换算法	165
3.5.1 向量处理方式	91	5.2.6 写策略	169
3.5.2 向量处理机的结构	93	5.2.7 Cache 性能分析	170
3.5.3 提高向量处理机性能的方法	95	5.2.8 改进 Cache 性能	173
3.5.4 向量处理机的性能评价	99	5.3 降低 Cache 不命中率	173
习题 3	103	5.3.1 三种类型的不命中	173
		5.3.2 增加 Cache 块大小	176
第 4 章 指令级并行	107	5.3.3 增加 Cache 的容量	177
4.1 指令级并行的概念	107	5.3.4 提高相联度	177
4.2 指令的动态调度	110	5.3.5 伪相联 Cache	178
4.2.1 动态调度的基本思想	110	5.3.6 硬件预取	179
4.2.2 Tomasulo 算法	112	5.3.7 编译器控制的预取	179
4.3 动态分支预测技术	124	5.3.8 编译优化	180
4.3.1 采用分支历史表 BHT	125	5.3.9 “牺牲”Cache	183
4.3.2 采用分支目标缓冲器 BTB	126	5.4 减少 Cache 不命中开销	183
4.3.3 基于硬件的前瞻执行	128	5.4.1 采用两级 Cache	184
4.4 多指令流出技术	133	5.4.2 让读不命中优先于写	187
4.4.1 基于静态调度的多流出技术	134	5.4.3 写缓冲合并	187
4.4.2 基于动态调度的多流出技术	136	5.4.4 请求字处理技术	188
4.4.3 超长指令字技术	138	5.4.5 非阻塞 Cache 技术	189
4.4.4 多流出处理器受到的限制	140	5.5 减少命中时间	189
4.4.5 超流水线处理机	141	5.5.1 容量小、结构简单的 Cache	190
4.5 循环展开和指令调度	144	5.5.2 虚拟 Cache	190
4.5.1 循环展开和指令调度的基本方法	144	5.5.3 Cache 访问流水化	193
4.5.2 静态超标量处理机中的循环展开	148	5.5.4 踪迹 Cache	193
习题 4	150	5.5.5 Cache 优化技术总结	193
		5.6 并行主存系统	195
第 5 章 存储系统	153	5.6.1 单体多字存储器	196
5.1 存储系统的层次结构	153	5.6.2 多体交叉存储器	196
5.1.1 存储系统的层次结构	153	5.6.3 避免存储体冲突	201
5.1.2 存储系统的性能参数	155	5.7 虚拟存储器	203
5.1.3 三级存储系统	156	5.7.1 基本概念	203
5.1.4 存储层次的 4 个问题	157	5.7.2 快速地址转换技术	204
5.2 Cache 基本知识	158	5.7.3 页式虚拟存储器实例:64 位 Opteron 的 存储管理	205
5.2.1 基本结构和原理	158	5.8 实例:AMD Opteron 的存储器层次结构	207
5.2.2 映像规则	159		

习题 5	212	7.4 动态互连网络	260
第 6 章 输入/输出系统	216	7.4.1 总线网络	260
6.1 输入/输出系统的性能	216	7.4.2 交叉开关网络	261
6.2 输入/输出系统的可靠性、可用性和 可信性	217	7.4.3 多级互连网络	262
6.3 廉价磁盘冗余阵列 RAID	218	7.4.4 动态互连网络的比较	267
6.3.1 RAID 0	221	7.5 消息传递机制	268
6.3.2 RAID 1	221	7.5.1 消息寻径方案	268
6.3.3 RAID 2	222	7.5.2 死锁与虚拟通道	272
6.3.4 RAID 3	223	7.5.3 流控制策略	273
6.3.5 RAID 4	223	7.5.4 选播和广播寻径算法	276
6.3.6 RAID 5	225	习题 7	278
6.3.7 RAID 6	225	第 8 章 多处理机	280
6.3.8 RAID 10 与 RAID 01	226	8.1 引言	280
6.3.9 RAID 的实现与发展	227	8.1.1 并行计算机系统结构的分类	281
6.4 总线	227	8.1.2 存储器系统结构和通信机制	282
6.4.1 总线的设计	227	8.1.3 并行处理面临的挑战	284
6.4.2 总线标准和实例	229	8.2 对称式共享存储器系统结构	287
6.4.3 与 CPU 的连接	231	8.2.1 多处理机 Cache 一致性	287
6.5 通道处理机	232	8.2.2 实现一致性的基本方案	289
6.5.1 通道的作用和功能	232	8.2.3 监听协议的实现	291
6.5.2 通道的工作过程	233	8.3 分布式共享存储器系统结构	296
6.5.3 通道的种类	235	8.3.1 目录协议的基本思想	296
6.5.4 通道流量分析	237	8.3.2 目录协议实例	300
6.6 I/O 与操作系统	239	8.3.3 目录的 3 种结构	303
6.6.1 DMA 和虚拟存储器	240	8.4 同步	305
6.6.2 I/O 和 Cache 数据一致性	241	8.4.1 基本硬件原语	305
习题 6	242	8.4.2 用一致性实现锁	307
第 7 章 互连网络	245	8.4.3 同步性能问题	309
7.1 互连函数	245	8.5 同时多线程	312
7.1.1 互连函数的表示方法	245	8.5.1 将线程级并行转换为 ILP	313
7.1.2 几种基本的互连函数	246	8.5.2 同时多线程处理器的设计	315
7.2 互连网络的结构参数与性能指标	252	8.5.3 同时多线程的性能	316
7.2.1 互连网络的结构参数	252	8.6 大规模并行处理机	317
7.2.2 互连网络的性能指标	252	8.6.1 并行计算机系统结构	317
7.3 静态互连网络	253	8.6.2 大规模并行处理机	319
		8.7 多处理机实例 1:T1	323
		8.8 多处理机实例 2:Origin 2000	327

习题 8	334	10.1 多核系统结构的需求	347
		10.1.1 功耗与散热问题	348
第 9 章 机群系统	336	10.1.2 并行度问题	348
9.1 机群的基本结构	337	10.1.3 应用软件的问题	349
9.1.1 机群的硬件	337	10.2 多核系统结构	350
9.1.2 机群的软件	338	10.2.1 多核的组织架构	350
9.2 机群的特点	340	10.2.2 多核系统结构实例	352
9.3 机群的分类	341	10.3 基于多核的并行程序设计	356
9.4 典型机群系统简介	342	10.3.1 并行编程模型	356
9.4.1 Berkeley NOW	342	10.3.2 并行语言	357
9.4.2 Beowulf	343	10.3.3 并行算法	358
9.4.3 LAMP	343	10.4 多核编程实例	359
9.4.4 IBM SP2	344	习题 10	361
习题 9	346		
第 10 章 多核系统结构与编程	347	参考文献	362

第 1 章 计算机系统结构的基本概念

主要内容

- 计算机系统的层次结构
- 计算机系统结构、计算机组成和计算机实现的概念
- 计算机系统结构的分类
- 定量分析技术
- 计算机系统结构的发展
- 计算机系统结构中并行性的发展

1.1 引言

第一台通用电子计算机诞生于 1946 年。60 多年来,计算机技术得到了飞速的发展,其速度之快,实在是令人赞叹。今天,用不到三四千元人民币购买的个人计算机,其性能、主存容量和硬盘容量都已经超过了 20 年前 100 万美元购买的计算机。对于许多应用来说,现在的高性能微处理器的性能已经超过了 10 多年前的超级计算机。这种惊人的发展一方面得益于计算机制造技术的发展,另一方面则是因为计算机系统结构的创新。

在计算机发展的开始 25 年中,上述两种因素都起着主要的作用,它们使计算机的性能每年提高约 25%。从 20 世纪 70 年代末到 80 年代初,大规模集成电路和微处理器的出现使计算机性能的年增长速度提高到了约 35%。这种提高主要归功于以集成电路为代表的制造技术的发展。由于大批量生产的微处理器在价格上有明显的优势,而且其性能也在不断地提高,所以基于微处理器的计算机占领了越来越多的市场份额。

到了 20 世纪 80 年代初,一种称为 RISC(Reduced Instruction Set Computer,精简指令集计算机)结构的出现,使计算机系统结构产生了一次重大变革。这种结构简化了指令系统,而把更多的芯片硅面积用于实现流水和 Cache,有效地提高了性能。从 20 世纪 80 年代中期以来,系统结构不断更新和变革,制造技术也不断发展,两者相结合,使计算机的性能增长达到了每年 50% 以上。这种情况维持了约 16 年。

这种飞速增长使得基于微处理器的系统在整个计算机设计领域占据了统治地位。服务器和个人计算机已经成为计算机工业的主要产品。小型机被用微处理器构建的服务器所取代,大型机也几乎已经被多处理机取代。这些多处理机是用若干个通用的微处理器构建的。甚至高端的超级计算机也是用许多微处理器来构建了。

但是,自2002年以后,计算机性能的年增长率下降到了约20%。其主要原因有3个:功耗问题(已经很大);可以进一步有效开发的指令级并行性已经很少;存储器访问速度的提高缓慢。Intel于2004年宣布放弃了其高性能单处理器项目,转向多核(multi-core)的研究和开发。这是通过在单个芯片上实现多个处理器来提高性能。IBM、SUN(已被甲骨文收购)、AMD等公司都纷纷采用了这一途径。这标志着系统结构的重大转折:从单纯依靠指令级并行转向开发线程级并行和数据级并行。

由此可见,计算机系统结构在计算机的发展中有着极其重要的作用。

1.2 计算机系统结构的概念

1.2.1 计算机系统的层次结构

现代计算机是由硬件/固件和软件组成的十分复杂的系统。为了对这个系统进行描述、分析、设计和使用,人们从不同的角度提出了观察计算机的观点和方法。其中常用的一种方法是从计算机语言的角度出发,把计算机系统按功能划分成多级层次结构,每一层以一种语言为特征,如图1.1所示。

一台计算机所提供的编程语言一般有多种,构成一系列的层次级别。这些语言按照从低级到高级的次序有微程序语言(仅供硬件设计者使用),机器语言,汇编语言,高级语言,应用语言(如SQL)等。对于用某一层语言来编写程序的程序员来说,一般不需关注其程序在计算机中是如何执行的,只要程序正确就能得到预期的结果。这样,对这层语言的程序员来说,似乎有了一种新的计算机,这层语言就是这种计算机的机器语言,该计算机能执行用这层语言编写的全部程序。

在图1.1中,L1是微程序机器级。这一级的机器语言是微指令集,其使用者是计算机硬件的设计人员,他们用微指令编写的微程序直接由固件/硬件来解释实现。



图 1.1 计算机系统的多级层次结构

L2 是传统机器级。这一级的机器语言就是传统的机器指令集。程序员用该指令集编写的程序由 L1 级上的微程序进行解释执行。

由微程序解释指令集又称为仿真(emulation)。实际上,在 L1 级上可以有多个能够在它上面运行的解释程序,每一个解释程序定义了一种指令集。因此,可以通过仿真在一台计算机上实现多种指令集。

有的计算机中没有采用微程序技术,因此没有微程序机器级(L1)。这时,L2 的指令集是由硬连逻辑直接解释执行的。硬连逻辑的优点是速度快。RISC 处理器经常采用这种实现(因为其指令集比较简单,详见 2.3.2 小节)。

L3 级是操作系统虚拟机。所谓虚拟机(virtual machine),是指由软件实现的机器,以区别于由固件/硬件实现的物理机器。这一级的机器语言由两部分构成,一部分是传统机器级指令,另一部分是操作系统级指令。后者用于实现对操作系统功能的调用,如打开/关闭文件、读/写文件等。用这一级语言编写的程序是由 L3 和 L2 来共同执行的,其中只有操作系统级指令是由操作系统进行解释执行。

L4 级是汇编语言虚拟机。这一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 L3 级和 L2 级语言,然后再由相应的计算机执行。完成这个翻译的程序称为汇编程序。

L5 级是高级语言虚拟机。这级的机器语言就是各种高级语言,目前高级语言已有上百种,常用的有 C/C++、Pascal、FORTRAN 等。用高级语言编写的程序一般由编译器翻译到 L4 或 L3 上。个别高级语言也用解释的方法实现,如绝大多数 BASIC 语言。

L6 级是应用语言虚拟机。这一级是为使计算机满足某种用途而专门设计的,因此这一级的语言就是各种面向具体应用问题的应用语言。可以设计专门用于人工智能、教育、行政管理、计算机设计等领域的虚拟机。用应用语言编写的程序一般是由应用程序包翻译到 L5 级上。

在上述 6 级层次结构中,通常 L1 ~ L3 级是用解释(interpretation)的方法实现,而 L4 ~ L6 级则是用翻译(translation)的方法实现。

翻译和解释是语言实现的两种基本技术。其共同点都是以执行一串 L 级指令来实现 L+1 级指令。其不同点在于:翻译技术是先把 L+1 级程序全部转换成 L 级程序后,再去执行新产生的 L 级程序,在执行过程中 L+1 级程序不再被访问;而解释技术则是每当一条 L+1 级指令被译码后,就直接去执行一串等效的 L 级指令,然后再去取下一条 L+1 级的指令,依此重复进行。这两种技术都被广泛使用。一般来说,解释执行比编译后再执行所花的时间多,但占用的存储空间较少。

1.2.2 计算机系统结构的定义

计算机系统结构(computer architecture)的经典定义是 1964 年 Amdahl 在介绍 IBM 360 系统时提出的:计算机系统结构是程序员所看到的计算机属性,即概念性结构与功

能特性。

按照计算机系统的多级层次结构,不同级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主要属性是该机器指令集的功能特性,而高级语言虚拟机程序员所看到的主要属性则是该计算机所配置的高级语言所具有的功能特性。

显然,不同的计算机系统,从传统机器级或汇编语言程序员的角度来看,具有不同的属性。但是从高级语言(如C语言)程序员的角度来看,它们几乎没有什么差别,具有相同的属性。换句话说,这些传统机器级所存在的差别对于高级语言程序员来讲是“看不见”的,也是不需要知道的。在计算机技术中,把这种本来存在的事物或属性,但从某种角度看又好像不存在的概念称为透明性(transparency)。通常,在一个计算机系统中,低层计算机的属性对高层计算机的程序员往往是透明的。可以看出,在计算机层次结构的各级上都有其系统结构。Amdahl提出的系统结构实际上是指传统机器语言级程序员所看到的计算机属性。

在J. L. Hennessy和D. A. Patterson编写的《Computer Architecture: A Quantitative Approach》一书中,把系统结构定义为囊括计算机设计的3个方面:指令集结构、组成和硬件。不妨将之理解为广义的系统结构定义。

在Amdahl的传统定义中,系统结构所包含的属性是指机器语言程序设计员(或者编译程序代码生成系统)为使其设计(或生成)的程序能在计算机上正确运行所需遵循的计算机属性。对于通用寄存器型计算机来说,这些属性主要是指:

- 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)。
- 数据表示(硬件能直接识别和处理的数据类型)。
- 寻址规则(包括最小寻址单元、寻址方式及其表示)。
- 寄存器定义(包括各种寄存器的定义、数量和使用方式)。
- 中断系统(中断的类型和中断响应硬件的功能等)。
- 机器工作状态的定义和切换(如管态和目态等)。
- 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连接方式、处理机/存储器与I/O设备间数据传送的方式和格式以及I/O操作的状态等)。

以上属性是计算机系统中由硬件或固件完成的功能,程序员在了解这些属性后才能编写出在传统机器级上正确运行的程序。因此,计算机系统结构概念的实质是确定计算机系统中软、硬件的界面,界面之上是软件实现的功能,界面之下是硬件和固件实现的功能。

1.2.3 计算机组成和计算机实现

计算机系统结构指的是计算机系统的软、硬件的界面,即机器语言程序员所看到的

传统机器级所具有的属性。

计算机组成(computer organization)指的是计算机系统结构的逻辑实现,包含物理机器级中的数据流和控制流的组成以及逻辑设计等。它着眼于物理机器级内各事件的排序方式与控制方式、各部件的功能以及各部件之间的联系。

计算机实现(computer implementation)指的是计算机组成的物理实现,包括处理机和主存等部件的物理结构,器件的集成度和速度,模块、插件、底板的划分与连接,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。

下面举例说明上述3个概念的区别。

① 确定指令集中是否有乘法指令属于计算机系统结构的内容,但乘法指令是用专门的乘法器实现,还是利用加法器经多步操作来实现,属于计算机组成。而乘法器、加法器的物理实现,如器件的选定及所用的微组装技术等,属于计算机实现。

② 主存容量与编址方式(按位、按字节或按字访问等)的确定属于计算机系统结构。为了达到给定的性能价格比,主存速度应多快、逻辑结构是否采用多体交叉等属于计算机组成。而主存系统的物理实现,如器件的选定、逻辑电路的设计和微组装技术的使用等均属于计算机实现。

可以看出,具有相同系统结构的计算机因为速度、价格等方面要求的不同,可以采用不同的计算机组成。而同一种计算机组成可以采用多种不同的计算机实现。系列机的出现充分反映了这一点。系列机(family machine)是指由同一厂家生产的具有相同系统结构、但具有不同组成和实现的一系列不同型号的机器。例如,IBM公司的IBM 370系列,Intel公司的x86系列等。1.4.2小节将进一步介绍系列机。

1.2.4 计算机系统结构的分类

常见的计算机系统结构分类法有两种:冯氏分类法和Flynn分类法。

冯氏分类法是用系统的最大并行度对计算机进行分类。它是由冯泽云先生于1972年提出的。最大并行度定义为:计算机系统在单位时间内能够处理的最大的二进制位数。可以用平面直角坐标系中的一个点代表一个计算机系统,其横坐标表示字宽(n 位),纵坐标表示一次能同时处理的字数(m 字)。 $m \times n$ 就表示了其最大并行度。

Flynn分类法是按照指令流和数据流的多倍性进行分类,它是M. J. Flynn于1966年提出的。Flynn分类法中定义:

- 指令流(instruction stream),即计算机执行的指令序列。
- 数据流(data stream),即由指令流调用的数据序列。
- 多倍性(multiplicity),即在系统受限的部件上,同时处于同一执行阶段的指令或数据的最大数目。

Flynn分类法把计算机系统的结构分为以下4类:

- ① 单指令流单数据流(Single Instruction stream Single Data stream, SISD)。