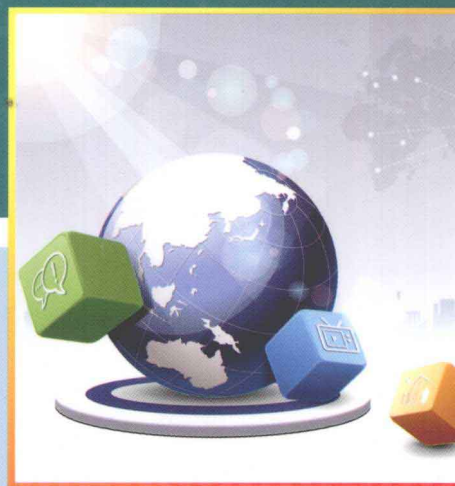


普通高等学校计算机科学与技术专业规划教材

计算机系统结构

COMPUTER ARCHITECTURE

徐洁 主编

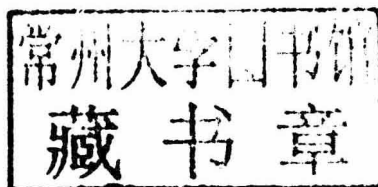


中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等学校计算机

计算机系统结构

徐 洁 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书根据“China Computing Curricula”和 ACM/IEEE-CS 教学计划,在深入研究国内外相关最新教科书和其他资料的基础上编写而成。

本书内容实用,实例丰富,既注重阐述现代微处理机所采用的设计技术,也分析了传统计算机系统的基本原理,知识点覆盖全面,内容有所取舍而不显庞杂,有利于读者学习和理解。

本书共分 8 章,包括概述、指令系统、流水线技术、指令级并行及限制、存储系统、输入/输出系统、多处理机系统、多计算机系统等,每章后附有习题,并免费提供电子课件。

本书适合作为高等学校计算机及相关专业的教材,也可作为相关领域技术人员的参考书。

图书在版编目(CIP)数据

计算机系统结构 / 徐洁主编. —北京:中国铁道出版社, 2012.1

普通高等学校计算机科学与技术专业规划教材
ISBN 978-7-113-14126-4

I. ①计… II. ①徐… 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(2012)第 003143 号

书 名: 计算机系统结构

作 者: 徐 洁 主编

策 划: 吴宏伟 杨 勇

读者热线: 400-668-0820

责任编辑: 孟 欣

编辑助理: 何 佳

封面设计: 付 巍

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.edusources.net>

印 刷: 三河市兴达印务有限公司

版 次: 2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 18.75 字数: 446 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-14126-4

定 价: 33.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 63549504

普通高等学校计算机科学与技术专业规划教材

编审委员会

主 任：蒋宗礼（北京工业大学）

副主任：王志英（国防科技大学）

杨 波（济南大学）

委 员：（按姓氏音序排列）

常会友（中山大学）

陈俊杰（太原理工大学）

陈 明（中国石油大学）

陈笑蓉（贵州大学）

陈志国（河南大学）

顾乃杰（中国科技大学）

胡 亮（吉林大学）

黄国兴（华东师范大学）

姜守旭（哈尔滨工业大学）

李仲麟（华南理工大学）

刘腾红（中南财经政法大学）

罗军舟（东南大学）

王国仁（东北大学）

王命延（南昌大学）

吴 跃（电子科技大学）

袁晓洁（南开大学）

岳丽华（中国科技大学）

张 莉（北京航空航天大学）

计算机学科虽然是一门年轻的学科，但它已经成为一门基础技术学科，在各个学科发展中扮演着重要的角色。因此，社会产生了对计算机科学与技术专业人才的巨大需求。目前，计算机科学与技术专业已成为我国理工专业中规模最大的专业，为高等教育发展做出了巨大贡献。近些年来，随着国家信息化建设的推进，作为核心技术的计算机技术，更是占有重要的地位。信息化建设不仅需要更先进、更便于使用的先进计算技术，同时也需要大批的建设人才。瞄准社会需求准确定位，培养计算机人才，是计算机科学与技术专业及其相关专业的历史使命，也是实现专业教育从劳动就业供给导向型向劳动就业需求导向型转变的关键，从而也就成为提高高等教育质量的关键。

教材在人才培养中占有重要地位，承担着“重要的责任”，这就确定了“教材必须高质量”这一基本要求。社会对计算机专业人才需求的多样性和特色，决定了教材建设的针对性，从而也造就了百花齐放、百家争鸣的局面。

关于高质量教材建设，教育部在提高本科教育质量的文件中都提出了明确要求。教高〔2005〕1号（2005年1月7日）文件指出，“加强教材建设，确保高质量教材进课堂。要大力锤炼精品教材，并把精品教材作为教材选用的主要目标。”“要健全、完善教材评审、评价和选用机制，严把教材质量关。”为了更好地落实教育部的这些要求，我们按照教育部高等学校计算机科学与技术教学指导委员会发布的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》所构建的计算机科学与技术专业本科教育的要求，组织了这套教材。

作为优秀教材的基础，我们首先坚持高标准，以对教育负责的精神去鼓励、发现、动员、选拔优秀作者，并且有意识地培育优秀作者。优秀作者保证了“理论准确到位，既有然，更有所以然；实践要求到位、指导到位”等要求的实现。

其次是按照人才培养的需要适当强调学科形态内容。粗略地讲：计算机科学的根本问题是“什么能被有效地自动计算”，科学型人才强调学科抽象和理论形态的内容；计算机系统工程的根本问题应该是“如何低成本、高效地实现自动计算”，工程型人才强调学科抽象和设计形态的内容；计算机应用的根本问题是“如何方便、有效地利用计算机系统进行计算”，应用型人才的培养偏重于技术层面的内容，强调学科设计形态的内容，在进一步开发基本计算机系统应用的层面上体现学科技术为主的特征。教材针对不同类型人才的培养，在满足基本知识要求的前提下，强调不同形态的内容。

第三是重视知识的载体作用，促进能力培养。在教材内容的组织上，体现大学教育的学科性和专业性特征，参考《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》示范性课程大纲，覆盖其要求的基本知识单元。叙述上力争引导读者进行深入分析，努力使读

者在知其然的基础上，探究其所以然。通过加强对练习和实践的引导，进一步培养学生的能力，促使相应课程在专业教育总目标的实现中发挥作用。

第四是瞄准教学需要，提供更多支持。近些年来，随着计算机技术、网络技术等在教学上的应用，教学手段、教学方式不断丰富，教材的立体化建设对丰富教学资源发挥了重要作用。通常，除主教材外，还要配套教学参考书、实验指导书、电子讲稿，有的还提供网络教学服务，等等。

第五是面向主要读者，强调教材的写作特征，努力做到叙述清晰易懂，语言流畅，深入浅出，有吸引力而不晦涩；追求描述的准确性，强调用词和描述的一致性，语言表达的清晰性和叙述的完整性；分散难点，循序渐进，防止多难点、多新概念的局部堆积。

我们相信，这套教材一定能够在培养社会需要的计算机专业人才上发挥重要作用，希望大家广为选用，并在使用中提出宝贵建议，使其内容不断丰富。

普通高等学校计算机科学与技术专业规划教材编审委员会

2008年1月

在当前的信息时代, 计算机技术的发展日新月异, 而引领计算机技术发展的是处理机(微处理机)的实现技术。近 20 年来, 随着超大规模集成电路 VLSI 技术的迅速发展, 处理机的系统结构也发生了深刻的变化。

曾经的大型计算机系统结构所采用的设计技术, 如今可在一块微处理机芯片上实现, 例如, 流水线结构、多个处理部件并发地执行指令操作、指令动态调度以及转移预测等。由于采用了这些设计技术, 微处理机的性能大幅提高。例如, 如果一个微处理机在每个时钟周期可以发出 4 条指令, 采用 5 级流水线, 则可能有多至 20 条指令同时在执行。

此外, 现代高性能的计算机系统已经普遍采用多处理机或多计算机的系统结构, 如多处理机的服务器系统、集群系统和大规模并行处理系统。而且计算机网络的迅速发展使得通信技术和计算机技术融合为一个不可分割的整体, 如集群系统是由一组完整的计算机通过高性能网络互连而成; 又如, 网格技术通过一套完整的网格中间件的支持, 将分布在互联网上的多种资源整合成一台巨大的超级计算机, 为使用者提供一套完善的、具有单一映像的支持环境。

现代的 VLSI 工艺仍在不断改进, 目前已经可以将多个微处理机制造在一块芯片上, 称为多核芯片, 当前的计算机和服务器已经逐渐采用多核芯片。

在这样的技术背景下, 计算机系统结构教材的内容也需要跟上计算机技术发展的步伐。编者在多年的教学实践和教材编写的基础上, 深入研究国内外先进教科书和相关的技术资料, 编写了这本与现代计算机技术吻合的教材。

本书主要涵盖了“China Computing Curricula”和 ACM/IEEE-CS 教学计划中的 CS-AR “计算机体系结构与组织”这一知识领域中的 6 个核心知识单元:

- AR4 存储系统组织和结构。
- AR5 接口和通信。
- AR6 功能组织。
- AR7 多处理和体系结构。
- AR8 性能提高技术。
- AR9 网络与分布式系统结构。

本书知识点覆盖全面, 内容更侧重于阐述现代实用微处理机所采用的主要设计技术, 并通过吸收国外先进教科书的内容, 尽量反映计算机系统设计的新技术和发展趋势。在阐述当前主流计算机系统采用的设计思想、结构原理、分析方法和性能评测的基础上, 分别在各章都给出了较新的实例产品进行配合说明。

本书编写层次分明, 结构清晰, 知识引入由浅入深。全书共分 8 章, 第 1、2 章介绍计算机系统结构的基础知识; 第 3~6 章阐述现代单 CPU 微处理机的主要设计技术; 第 7、8 章描述多处理机与多计算机系统的相关内容, 也介绍计算机网络与分布式计算技术发展融合产生的网格技术及实例。本书参考学时为 40~60 学时。其内容组织体现了下述的教学思路:

第 1 章通过回顾计算机系统的发展过程, 说明了为提高系统性能对系统结构进行的主要改进之处; 对有效提高系统性能的并行技术和并行计算机进行了讨论; 引入了现代计算机系统的

分类方法、设计方法以及性能评测的方法。第2章阐述了指令系统优化设计的方法和技术；深入讨论了指令系统发展中的CISC和RISC两种设计思想，强调现代RISC技术更注重支持流水线的高效率执行和编译器生成优化代码，对典型RISC处理机MIPS R4000和SPARC的指令系统进行了分析和比较。

由于单CPU微处理机设计技术是现代微处理机（包括多核芯片）设计的基础，因此本书在第3、4章深入分析和阐述了相关技术，即：流水线基础、MIPS R4000流水线、相关性对指令级并行的影响、支持指令级并行的编译技术、指令的静态与动态调度、静态与动态转移预测、多发射技术、指令级并行的支持与限制性因素、Intel Pentium 4实例分析。第5章对高速缓冲存储器（Cache）和虚拟存储器工作机制进行了深入分析，也讨论了主存储器的性能优化、进程保护和实例Alpha 21064。第6章从系统结构的角度介绍了输入/输出系统，包括输入/输出系统性能的测试与分析，以及磁盘冗余阵列（RAID）。

随着多核芯片的普及，计算机将普遍成为多处理机系统，因此在第7章深入分析了相关技术：多处理机系统的结构、多核处理机、多处理机互连网络、多处理机系统控制、多处理机系统性能分析、实例CRAY T3E与SGI Origin 2000，并介绍了并行处理语言与算法。第8章主要讨论两类典型多计算机系统：集群系统与MPP系统；介绍了集群系统的结构、通信技术、资源管理调度、并行程序设计环境，实例NOW、Beowulf和Cluster 1350；然后介绍了MPP系统的结构，实例ASCI White、BlueGene/L、ASCI Option Red和天河一号；最后介绍了网格技术和网格体系结构，以及网格实例Globus。

本课程的先修课程是“计算机组成原理”、“汇编语言程序设计”、“数据结构”等课程，本课程可以在“操作系统”、“编译原理”课程之后或与它们同时开设。

本书由徐洁主编并编写第1、2章，吴晓华编写第3、4章，丁旭阳编写第5章，王雁东编写第6章，胡健编写第7、8章。吉林大学胡亮教授担任主审，并提出了许多宝贵的修改意见。本书编写过程中，还得到了电子科技大学计算机学院领导、老师和同学的热情支持。在此，谨向所有给予我们支持和帮助的同志表示衷心的感谢。

由于时间仓促，编者水平有限，书中难免存在疏漏与不足之处，恳请读者与同行给予批评指正。

编 者
2011年11月

第 1 章 概述	1
1.1 计算机系统结构的概念	1
1.1.1 计算机系统的层次结构	1
1.1.2 计算机系统结构、组成与实现	4
1.2 计算机系统结构的发展	7
1.2.1 冯·诺依曼计算机	8
1.2.2 存储程序计算机系统结构的发展过程	8
1.2.3 非冯·诺依曼结构计算机	10
1.3 并行性与并行计算机	11
1.3.1 并行性概念	11
1.3.2 提高并行性的技术途径	11
1.3.3 并行计算机简介	12
1.4 计算机系统的分类	16
1.4.1 Flynn 分类法	16
1.4.2 应用分类	17
1.5 计算机系统设计的主要任务与量化原则	19
1.5.1 计算机系统设计者的主要任务	19
1.5.2 计算机系统设计的量化原则	21
1.6 计算机系统的性能评测	24
1.6.1 计算机性能评估	25
1.6.2 计算机性能的测试	26
1.6.3 计算机性能测试结果的统计和比较	30
习题	32
第 2 章 指令系统	35
2.1 数据表示	35
2.1.1 浮点数表示	36
2.1.2 自定义数据表示	40
2.2 寻址技术	41
2.2.1 编址方式	42
2.2.2 寻址方式	43
2.2.3 程序在主存中的定位方法	46
2.3 指令格式的设计和优化	47
2.3.1 指令操作码的优化	47
2.3.2 地址码的优化表示	51
2.3.3 指令格式设计实例	53
2.4 指令系统的改进	56
2.4.1 RISC 与 CISC 的概述	56
2.4.2 RISC 指令系统实例	59
习题	67
第 3 章 流水线技术	69
3.1 流水线概述	69
3.1.1 流水线的基本概念	69
3.1.2 流水线的分类	73

3.1.3	流水线的特点	76
3.2	流水线的时空图及性能分析	77
3.2.1	流水线的时空图	77
3.2.2	流水线的性能分析	78
3.3	流水线中的相关	83
3.3.1	什么是流水线相关	83
3.3.2	流水线中的结构相关(资源相关)	84
3.3.3	流水线中的数据相关	86
3.3.4	流水线的控制相关	87
3.4	MIPS R4000 流水线计算机	89
3.4.1	MIPS R4000 流水线计算机基本结构和工作原理	89
3.4.2	MIPS R4000 流水线	92
3.5	向量处理机	94
3.5.1	向量处理的基本概念	94
3.5.2	向量处理机的结构	97
3.5.3	向量指令的执行过程及简单性能计算	98
3.5.4	向量的链接技术	100
3.5.5	提高向量处理机的方法	103
3.5.6	向量处理机的性能评价	105
习题		108
第 4 章	指令级并行及限制	111
4.1	指令级并行概述	111
4.1.1	指令级并行的基本概念	111
4.1.2	相关性对指令级并行的影响	112
4.1.3	支持指令级并行的基本编译技术	115
4.2	指令的动态调度	118
4.2.1	动态调度的原理	119
4.2.2	记分牌动态调度算法	119
4.2.3	Tomasulo 动态调度算法	126
4.3	转移预测技术	135
4.3.1	静态转移预测	135
4.3.2	动态转移预测	137
4.4	多发射技术	142
4.4.1	超标量技术	142
4.4.2	多发射的动态调度	145
4.4.3	超长指令字技术	151
4.5	指令级并行的支持与限制	152
4.5.1	窗口大小和最大发射数目对理想处理机的限制	153
4.5.2	实际分支和分支预测的影响对理想处理机的限制	156
4.5.3	有限数目寄存器的影响	158
4.5.4	非完美别名分析造成的影响	159
4.6	Intel Pentium 4 实例分析	161
4.6.1	Pentium III 和 Pentium 4 结构的简单比较	161
4.6.2	Pentium III 和 Pentium 4 性能的简单比较	162
习题		163
第 5 章	存储系统	165
5.1	存储系统简介	165
5.1.1	存储系统的层次结构	165
5.1.2	存储系统的性能参数	166

5.2	高速缓冲存储器 (Cache)	168
5.2.1	Cache 的工作原理	169
5.2.2	地址映像与变换方法	170
5.2.3	Cache 替换算法及实现	174
5.2.4	Cache 的一致性问题	178
5.2.5	Cache 的性能分析	179
5.3	Cache 性能的优化	181
5.3.1	降低 Cache 失效率的方法	181
5.3.2	减少 Cache 失效开销	183
5.3.3	减少命中时间	184
5.4	主存储器及性能优化	185
5.4.1	主存储器	185
5.4.2	性能优化	185
5.5	虚拟存储器	187
5.5.1	工作原理	187
5.5.2	地址映像与变换	192
5.5.3	内部地址变换优化	193
5.5.4	页面替换算法及实现	195
5.5.5	提高主存命中率的方法	198
5.6	进程保护和虚拟存储器实例	199
5.6.1	进程保护	200
5.6.2	Alpha 21064 存储管理	201
5.7	Alpha 21264 存储层次结构	203
	习题	206
第 6 章	输入/输出系统	208
6.1	输入/输出系统概述	208
6.1.1	输入/输出系统的特点	208
6.1.2	基本的输入/输出方式	210
6.2	总线	211
6.2.1	总线概述	211
6.2.2	总线的连接方式	212
6.3	通道处理机	213
6.3.1	通道的功能	213
6.3.2	通道的工作过程	214
6.3.3	通道的种类	215
6.3.4	通道中的数据传送过程	217
6.3.5	通道流量分析	218
6.4	外围处理机	220
6.4.1	输入/输出处理机的作用	220
6.4.2	输入/输出处理机的种类和组织形式	221
6.4.3	输入/输出处理机实例	221
6.5	I/O 系统性能评测	222
6.5.1	I/O 系统的可靠性、可用性和可信性	222
6.5.2	I/O 子系統性能衡量	223
6.5.3	I/O 子系统的设计	224
6.5.4	并行 I/O 基本原理	225
6.5.5	排队论简介	226
6.6	磁盘冗余阵列	228
6.6.1	RAID 概述	228

6.6.2 RAID 系统分级	228
习题	234
第 7 章 多处理机系统	237
7.1 多处理机系统结构	237
7.1.1 多处理机系统的硬件结构	237
7.1.2 多处理机系统的存储器组织形式	239
7.1.3 多处理机系统的操作系统	241
7.1.4 多核处理机	243
7.2 多处理机的互连网络	244
7.2.1 互连网络的基本概念	244
7.2.2 互连网络的特性	245
7.2.3 互连网络的类型	246
7.3 多处理机系统的系统控制	250
7.3.1 多处理机系统的调度	250
7.3.2 多处理机系统的进程通信	252
7.4 并行处理语言及算法	254
7.4.1 并行处理遇到的挑战	254
7.4.2 并行编程模型	255
7.4.3 并行语言	256
7.4.4 并行算法	257
7.5 多处理机的性能	260
7.5.1 任务粒度	260
7.5.2 基本模型	260
7.5.3 通信开销线性增加的模型	262
7.5.4 完全重叠通信的模型	262
7.5.5 具有多条通信链的模型	262
7.6 多处理机系统实例	263
7.6.1 CRAY T3E 系统	263
7.6.2 SGI Origin 2000 系列服务器	264
习题	266
第 8 章 多计算机系统	267
8.1 集群计算机系统	267
8.1.1 集群系统的基本概念和结构	267
8.1.2 集群系统的特点	268
8.1.3 集群系统的通信技术	269
8.1.4 集群系统资源管理和调度	271
8.1.5 集群系统并行程序设计环境	272
8.2 典型集群系统实例	273
8.3 MPP 系统	276
8.3.1 MPP 系统机构	276
8.3.2 基于 MPP 的并行计算机系统	277
8.3.3 集群系统与 MPP 系统	278
8.4 网格技术	278
8.4.1 网格基础	279
8.4.2 网格技术概念	279
8.4.3 网格体系结构	280
8.5 网格实例	283
习题	285
参考文献	287

第1章

概述

从低端的单片机到高端的并行计算机系统，其价格、体积大小、性能和应用千差万别，而且仍在不断发展变化中，但它们都始终采用了一些经典的基本概念，例如：计算机系统结构、组成与实现的定义、冯·诺依曼结构、系列机、并行性、Flynn 分类法等，这些概念将在本章分别阐述。

从计算机系统发展 60 多年的历史来看，计算机系统性能的不断提高主要依靠器件技术的迅速发展以及系统结构的改进。本章将回顾计算机系统的发展过程，并着重说明系统结构的主要改进之处。

并行技术一直都是提高计算机系统性能的有效手段，对计算机系统结构的改进起着重要的作用。此外，采用什么方式对现有计算机系统的性能进行客观公正的评测，对计算机的设计者与使用者来说也是尤其重要的。本章将介绍并行性概念与并行计算机，最后讨论计算机系统的性能评测方法。

现代计算机系统的复杂性对设计者提出了更高的要求，本章还将探讨设计者的主要任务、计算机系统设计的经验方法，以及用于设计方案评价的重要公式 Amdahl 定律和 CPU 性能公式。

1.1 计算机系统结构的概念

本节先从不同的角度讨论计算机系统的层次结构，然后介绍计算机系统结构、组成与实现的定义以及相互关系，同时也介绍相关的常用术语。

1.1.1 计算机系统的层次结构

计算机系统是由软件和硬件组成的复杂系统，因此在分析设计时通常采用层次结构的观点和方法去描述系统的组成与功能，这样可以控制计算机的复杂性，并使得计算机的分析与设计可以有组织地、系统地进行。下面将分别从系统内部的有机组成和程序设计语言功能的角度，介绍两种常用的层次结构模型。

1. 从计算机系统组成角度划分层次结构

如果从计算机系统组成的角度来划分层次，一般计算机系统都包含 5 层，如图 1-1 所示。

图 1-1 自下而上地描述了构成计算机系统的硬件层和多个软件层，直观展示了构建一台计算机时的逐层生成过程。

(1) 第 1 层 微体系结构层是具体存在的硬件层次，它执行机器指令，可看做指令系统层指令的解释器。在由微程序控制的计算机上，微程序就是上一层指令的解释器。而在硬件直接控制的计算机上，是由硬件直接解释执行指令，并不存在一个真正的程序来解释上一层的指令。

(2) 第 2 层 指令系统层是机器语言程序员所看到的计算机，这一层也称为传统机器级。它是一个抽象的层次，其主要特征就是指令系统。指令系统层定义了硬件与编译器之间的接口，它是一种硬件和编译器都能理解的语言。一方面，指令系统会表明一台计算机具有哪些硬件功能，是硬件逻辑设计的基础。因此，在指令系统层，应该定义一套在当前和将来的技术条件下能够高效率实现的指令集。另一方面，指令系统层需要为编译器提供明确的编译目标，使编译结果具有规律性和完整性。

(3) 第 3 层 操作系统层。从系统程序员的角度来看，操作系统是一个在指令系统层提供的指令和特性之上又增加了系统调用和特性的程序。这一层增加的系统调用是由运行在指令系统层上的操作系统解释执行的。操作系统层并不是为普通程序员的使用而设计的，它主要是为支持高层所需的解释器或编译器的运行而设计的。

(4) 第 4 层 汇编语言层。由于直接用机器指令代码编程非常困难，汇编语言实际上就是“符号化”的机器语言，每一条机器语言指令都有一条汇编指令语句与之对应，它是面向机器结构的语言。用汇编语言编写的程序先由汇编器翻译成机器语言程序，然后由微体系结构层解释执行。

从这一层看去，每一种计算机都有一套自己的汇编语言，解释它的汇编器，以及相应的程序设计与开发方法。汇编语言层以及上层是提供给解决应用问题的应用程序员使用的。

(5) 第 5 层 面向问题语言层。该层的语言通常是为解决现实问题的应用程序员使用的，这些语言称为高级语言。目前，已开发的高级语言有几百种，如 C、C++、Java、Lisp 和 Prolog 等，用这些语言编写的程序一般会先由编译器翻译成第 3 层或第 4 层能识别的语言，然后再执行，偶尔也有解释执行的。

在计算机系统中，第 1、2、3 层结构使用的机器语言指令用二进制代码表示，虽然适合机器执行，但不容易被人理解。从第 4 层开始，所使用的语言是人们能理解的单词和缩略语。因此，在第 2 层和第 3 层主要用解释方式执行指令；而第 4 层和第 5 层通常采用编译方式执行指令。

总之，用分层方式来设计和分析计算机系统时，设计人员可以忽略一些无关紧要的细节，让系统设计中复杂的问题变得更容易理解。

2. 从语言功能角度划分层次结构

如果将计算机功能抽象为“能执行用某些程序设计语言编写的程序”，那么我们看到的就是图 1-2 所示的一种语言功能层次模型。计算机硬件的物理功能只是执行机器语言，称为机器

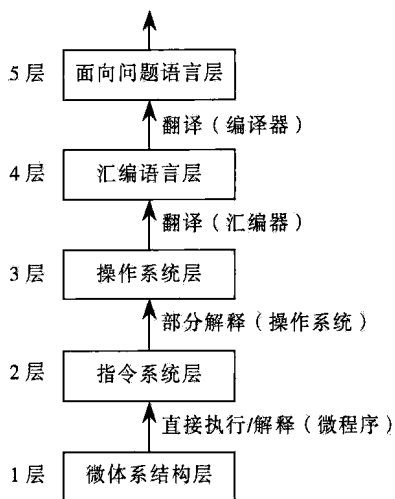


图 1-1 从计算机系统组成角度划分的层次结构模型

语言物理机（当然，还可细分出一个微程序级）。从这一级我们看到的是一台实际的机器。

一般情况下，计算机都是先将用程序设计语言编写的程序翻译成机器语言，然后才能由物理机执行。但当高级语言较复杂时，也可分级编译，即将高级语言先翻译成层次低些的某种中间语言，再将中间语言进一步翻译成机器语言，如图 1-2 中虚线所示。

如图 1-2 所示，从使用某种高级语言编程的程序员的角度来看计算机，看到的就可以执行这种高级语言的机器，即具有这种高级语言功能的虚拟机。虚拟机是指通过配置软件（如某种语言的编译器或解释器）扩充机器功能后所形成的一台计算机，实际机器并不具备这种功能。例如，C 语言程序员看到计算机能接收并执行 C 语言编写的程序，从他的角度看到的就是一台可以执行 C 语言的虚拟机，然而实际上物理机只能执行其机器语言，它是通过配置 C 语言编译程序才能处理 C 语言程序的。

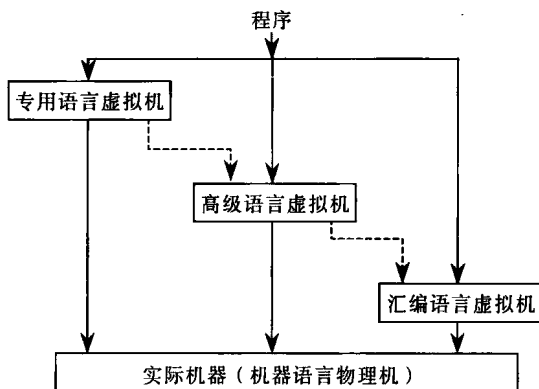


图 1-2 从语言功能角度划分的层次结构模型

采用虚拟机概念是计算机设计中的又一重要策略，它将提供给用户的功能抽象出来，使其脱离具体的物理机器，这有利于让用户摆脱真实物理机细节的束缚，获得超越物理机的功能。例如，为了使 Java 程序能在不同的计算机上运行，SUN 公司（已于 2009 年 Oracle 公司收购）定义了一种称为 Java 虚拟机（Java Virtual Machine, JVM）的虚拟体系结构。它有 32 位字组成的内存，能执行 226 条指令，大多数指令都很简单，只有少量较复杂的指令。

小知识

SUN 公司不仅提供了一个将 Java 语言程序编译成 Java 虚拟机指令序列（又称 Java 执行程序）的编译器，以实现程序的跨平台运行，而且还实现了能解释执行 Java 程序的解释器，该解释器用 C 语言编制，可在任何一台有 C 编译器的计算机上运行。例如，通常 Internet 的浏览器中就包含了 JVM 解释器，用来运行网页中的小 Java 执行程序 Applet（为网页提供语音和动画）。但是，这种解释器的执行速度较慢，因此产生了另一种运行方法，即在浏览器中包含一个将 JVM 程序翻译成机器语言的编译器，并在需要时激活它，这种随时编译的编译器称为 JIT（Just In Time）。这种运行方式的问题是第一次运行时的开销较大，即从收到 JVM 程序到开始执行会有因编译产生的延时，然而对于重复使用的 JVM 程序来说该方式（编译方式）的效果更佳。

除了软件实现的 JVM 机（JVM 解释器和 JIT 编译器）之外，SUN 和其他一些公司还设计了硬件的 JVM 芯片，即设计出可直接执行 JVM 程序的 CPU，不再需要用一层软件来解释或用 JIT 来编译 JVM 程序。这种体系结构的芯片 picoJava-I 和 picoJava-II 已经出现在嵌入式系统市场上。

3. 软、硬件在逻辑上的等价

计算机系统以硬件为基础，通过软件扩充其功能，并以执行程序方式体现其功能。一般来说，硬件只完成基本的功能，而复杂的功能则通过软件来实现。但是，硬件与软件之间的界面，如功能分配关系，常随技术发展而变化。有许多功能既可以直接由硬件实现，也可以

在硬件支持下靠软件实现,对用户来说在功能上是等价的,我们称为软、硬件在功能上的逻辑等价。例如,乘法运算,可由硬件乘法器实现,也可在加法器与移位器支持下由乘法子程序实现。又如,前述的 Java 虚拟机既可以用软件(JVM 解释器或 JIT 编译器)实现,也可以用硬件(JVM 芯片)实现。

从设计者角度看,指令系统是硬件与软件之间的界面。硬件的基本任务是识别与执行指令代码,因此指令系统所规定的功能一般可由硬件实现。用程序设计语言编制的程序最终需要转换成指令序列,由指令代码表示才能执行,因此指令系统是编程的基础(直接或间接)。如何设计指令系统,选择恰当的软、硬件功能,取决于所选定的设计目标、系统的性能价格比等因素,并与当时的技术水平有关。

刚出现数字计算机时,人们依靠硬件实现各种基本功能。但是,由于硬件造价高,随后设计计算机系统时就采用了硬件软化的技术策略,即只让硬件实现较简单的指令系统,如加、减、移位与基本逻辑运算功能,而依靠软件实现乘、除、浮点运算等更复杂的功能。曾经的小型计算机就采用了这种技术策略,从而使得它们的造价不高、结构简单,同时又具有较强的功能。

随着集成电路技术的飞速发展,我们可以在一块芯片上集成相当强的功能模块,于是计算机系统设计又出现了另一种技术策略——软件硬化,即将原来依靠软件才能实现的一些功能,改由大规模、超大规模集成电路直接实现,如浮点运算、存储管理等功能。这样,系统将具有更高的处理速度和更强的功能。

如果说系统设计者必须关心软、硬件之间的界面,即哪些功能由硬件实现,哪些由软件实现,用户则更关心系统究竟能提供哪些功能。至于这些功能是由硬件还是软件实现,在逻辑功能上则是等价的,只是执行速度有差别而已。

1.1.2 计算机系统结构、组成与实现

1. 基本概念

计算机系统的层次结构表明,不同层(级)的计算机具有不同的属性。传统机器级程序员所看到计算机的主要属性是该机指令系统的功能特性,而高级语言虚拟机程序员所看到计算机的主要属性是该机所配置的高级语言具有的功能特性。

对于不同的计算机系统,从传统机器级程序员或汇编语言程序员的角度来看,具有不同的属性。但是,从某一种高级语言程序员看,它们就几乎没有什么差别,具有相同的属性。因此,从高级语言程序员的角度,是“看不见”不同计算机系统在传统机器级上存在的差别的。在计算机技术中,对这种本来是存在的事物或属性,但从某种角度看又好像不存在的概念称为透明性(Transparency)。在一个计算机系统中,低层机器的属性对高层机器的程序员往往是透明的。

计算机系统结构,也称为计算机体系结构(Computer Architecture)。早在 1964 年 C.M.Amdahl 就给出了它的定义:计算机系统结构是机器语言程序员(或编译程序设计者)所看到的计算机的属性,是硬件子系统的概念性结构与功能特性。

Amdahl 提出的系统结构是指传统机器级的系统结构,是编译程序生成的机器语言目标程序能够在机器上正确运行应遵循的计算机属性。实际上,传统机器级系统结构的属性主要是由该机器的指令系统来表征的,具体属性如下:

- (1) 数据表示:硬件能直接识别和处理的数据类型。
- (2) 寄存器定义:包括各种寄存器的定义、数量和使用方式。

(3) 指令系统：寻址规则、机器指令的操作类型和格式、指令间的排序和控制机制。

(4) 中断系统：中断的类型和中断响应硬件的功能等。

(5) 机器工作状态的定义和切换：如管态和目态等。

(6) 存储系统：主存最小编址单位、编址方式、可编程最大存储容量等。

(7) 输入/输出结构：输入/输出连接方式、处理机存储器与输入/输出设备间数据传送的方式和格式、输入/输出操作的状态等。

(8) 信息保护：信息保护方式和硬件对信息保护的支持。

这些属性表明，经典计算机体系结构概念的实质就是计算机系统中软硬件界面的确定，界面之上是软件的功能，界面之下是硬件的功能。显然，经典计算机系统结构定义的范畴不包括机器级内部数据流和控制流的组织，也不包括逻辑设计与物理实现。

计算机组成 (Computer Organization)，也称为计算机组织。在计算机系统结构确定分配给硬件子系统的功能及其概念结构之后，计算机组成的任务是研究硬件子系统各组成部分的内部结构和相互联系，以实现机器指令级的各种功能和特性。它包括：数据通路宽度的确定，各种功能部件的相互连接及性能参数的匹配，功能部件的并行性确定，控制部件的设计，缓冲器和排队的使用，可靠性技术的采用等。例如，AMD Opteron 64 与 Intel Pentium 4 的指令系统相同，即两者的系统结构相同；但内部组成不同，流水线和 Cache 结构是完全不同的，相同的程序在两个机器上的运行时间一般是不同的。

计算机实现 (Computer Implementation)，是指计算机组成的物理实现。它包括处理机、主存储器、I/O 子系统等部件的物理结构，器件的集成度和速度，信号传输，器件、模块、插件、底板的划分与连接，专用器件的设计，电源、冷却、装配等技术以及有关的制造技术和工艺等。例如，同一系列机中的 Pentium 4 与移动版 Pentium 4 具有相同的指令系统和基本相同的组成，但由于是不同档次的机器，其硬件实现是不同的，两者的时钟频率和存储系统是不同的。

计算机系统结构、计算机组成和计算机实现是 3 个不同的概念。计算机系统结构是指令系统及其模型；计算机组成是计算机系统结构的逻辑实现；计算机实现是计算机组成的物理实现。它们各自包含不同的内容和采用不同的技术，但又有紧密的关系。

随着计算机技术的迅速发展，现代计算机系统设计面临的问题与 10 年前也大不相同。计算机系统结构、组成与实现之间的界限变得越来越模糊。现在使用的是广义的计算机体系结构概念，它既包括经典的计算机体系结构的概念范畴，也包括对计算机组成和计算机实现技术的研究。

当前计算机系统设计的任务异常复杂，涉及很多领域的技术，从编译程序、操作系统到指令系统设计、功能结构设计、逻辑设计和实现技术，特别是实现技术的发展对系统结构具有深远的影响，这些实现技术包括集成电路、半导体 DRAM、磁盘技术和网络实现技术。

通常，计算机系统设计的首要任务是要明确功能需求，即机器的应用领域，如设计目标是通用桌面机还是科学计算桌面机。也包括适应市场的需求，如果市场已存在为某一指令系统设计的大量软件，系统结构设计者应该考虑在新的机器中与该指令系统兼容。同时，还要考虑支持选定的操作系统所必需的特性，如存储管理和信息保护。对硬软件技术标准的支持也是很重要的，如浮点标准、典型 I/O 标准、操作系统、网络和编程语言等。

2. 系列机

系列机 (Family Machine) 是指由一个制造商生产的具有相同的系统结构，但具有不同