高等学校教材

计算机系统结构

（第二版）

李学干　苏东庄

西安电子科技大学出版社

(陕)新登字 010 号

内 容 简 介

本书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的一些进展。

全书共八章。第一章讲述计算机系统层次结构，计算机系统结构、组成、实现的定义和相互关系，软硬取舍原则及设计方法，软件移植手段，应用与器件的影响，并行性发展与计算机系统分类。第二章讲述数据表示，寻址方式，指令系统的设计与改进，RISC 技术。第三章讲述总线，中断系统，通道处理机和外围处理机。第四章讲述存贮体系，虚拟存贮器，Cache 存贮器，主存保护。第五章讲述重叠，流水和向量处理机。第六章讲述并行处理机和相联处理机。第七章讲述多处理机。第八章讲述面向对象的计算机，数据流机，归约机和智能机。

本书内容丰富，每章均有大量习题，可作为计算机专业本科生和有关专业研究生的教材，也可作为科技人员的参考书。

高等学校教材

计算机系统结构(第二版)

李学干　苏东庄

责任编辑　谭玉瓦

西安电子科技大学出版社出版

西安电子科技大学印刷厂印刷

陕西省新华书店发行　各地新华书店经售

开本 787×1092　1/16　印张 22.8/16　字数 532 千字

1984 年 11 月第 1 版　1991 年 6 月第 2 版　1995 年 10 月第 10 次印刷　印数 112 001—122 000

ISBN 7-5606-0139-1/TP·0046(课)　　定价：16.00 元

出 版 说 明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校工科电子类专业课教材的编审、出版的组织工作。从1977年底到1982年初，由于各有关院校，特别是参与编审工作的广大教师的努力和有关出版社的紧密配合，共编审出版了教材159种。

为了使工科电子类专业教材能更好地适应社会主义现代化建设培养人才的需要，反映国内外电子科学技术水平，达到“打好基础、精选内容、逐步更新、利于教学”的要求，在总结第一轮教材编审出版工作经验的基础上，电子工业部于1982年先后成立了高等学校《无线电技术与信息系统》、《电磁场与微波技术》、《电子材料与固体器件》、《电子物理与器件》、《电子机械》、《计算机与自动控制》，中等专业学校《电子类专业》、《电子机械类专业》共八个教材编审委员会，作为教材工作方面的一个经常性的业务指导机构，并制定了1982～1985年教材编审出版规划，列入规划的教材、教学参考书、实验指导书等共217种选题。在努力提高教材质量，适当增加教材品种的思想指导下，这一批教材的编审工作由编审委员会直接组织进行。

这一批教材的书稿，主要是从通过教学实践、师生反映较好的讲义中评选择优和从第一轮较好的教材中修编产生出来的。广大编审者，各编审委员会和有关出版社都为保证和提高教材质量作出了努力。

这一批教材，分别由电子工业出版社、国防工业出版社、上海科学技术出版社、西北电讯工程学院出版社、湖南科学技术出版社、江苏科学技术出版社、黑龙江科学技术出版社和天津科学技术出版社承担出版工作。

限于水平和经验，这一批教材的编审出版工作肯定还会有许多缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评建议，共同为提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

第二版前言

《计算机系统结构》是高等院校工科电子类计算机专业统编教材之一，由计算机与自动控制教材编审委员会计算机编审小组评选审定，并推荐出版。经过多年的使用，由于计算机的飞速发展，我们觉得有必要作一次较大幅度的修改。许多院校的同志们也提出了这样的希望。

在这次重新编写后，变动较大的有如下几点：我们考虑到许多学校已设置有专门的"计算机可靠性技术"和"计算机系统性能与评价"课程，而且已出版了专门的教材，所以这部分内容就不再列入本教材中；新增加了有关RISC技术、外围处理机、相联处理机、基于面向对象程序设计的计算机、数据流机、归约机、智能机等内容；适当扩充向量处理机、并行处理机和多处理机的部分内容；对原有章节内容根据情况重新作了取舍、补充和内容更新，并对全书各章的内容编排上作了较大调整。

本教材仍按"研究软、硬件功能分配以及如何最佳、最合理地实现分配给硬件的功能"这个方向来编写，仍然着眼于叙述基本概念、基本原理、基本结构和基本分析方法。虽然列举了不少实际机器的例子，但并不是围绕这些机型或过多地从具体实现上的细节来讲述。本教材力求反映出近十几年来在系统结构上的重要进展和今后可能的发展。

本课程应在"计算机(组成)原理"、"程序设计语言"和"数字逻辑"等课程之后开设。学生最好有"数据结构"方面的知识。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。本课程也可作为其它相关专业的研究生或本科生的选修课程。本课程的参考教学时数为100～120学时。

本书在编写过程中多次得到有关领导部门及不少兄弟院校、研究所的专家、教授和同志们的热情鼓励和帮助，有的曾提出过宝贵的意见。西安电子科技大学出版社的同志们为本书的出版也做了大量工作，我们在此表示衷心的感谢。

应当说明的是，考虑到教材的继承性，不可能对原有教材在体系、内容和风格上作太大的变动，恳切希望读者对本书的缺点和错误予以指正。

编著者

1990年7月

第一版前言

本教材是高等院校工科电子类计算机专业统编教材之一，由计算机与自动控制教材编审委员会计算机编审小组评选审定，并推荐出版。

本教材是按计算机编审小组审定的编写大纲进行编写和审阅的，由西北电讯工程学院苏东庄主编，清华大学薛宏熙主审。

本教材是按"研究软、硬件功能分配以及如何最佳、最合理地实现分配给硬件的功能"这个方向来编写的，着重于基本概念、基本原理、基本结构和分析方法。虽然在讲述时联系了实际机器，但并不是围绕某种机型或过多地从具体实现上的细节来讲述。本教材力求反映近十几年来在系统结构上的重要进展以及今后可能的发展。

本课程应在"计算机(组成)原理""程序设计语言"和"数字逻辑"等课程之后开设。学生最好有"数据结构"方面的知识。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。本书的第六、七、八章是按学生不选修"容错与诊断"、"纠错码"、"并行处理计算机结构"和"计算机系统性能评价"等课来写的。本课程的参考教学时数为80～100学时。

本教材是在国防工业出版社1981年出版的《计算机系统结构》的基础上修编而成的。参加当时编写的有：西北电讯工程学院的苏东庄、张志华、李学干、马玉珍；清华大学的金兰；华东师范大学的张东韩和唐培艺，由苏东庄主编。担任主审的是清华大学的房家国和薛宏熙。参加此次修编的除苏东庄外，还有李学干、金兰(第七章)、梁新来(第六章)、马玉珍(第八章)，李学干承担了全书的文稿整理工作。

这里特别要感谢本版主审人薛宏熙同志，他进行了认真细致的审稿，并提出了宝贵的修改意见。凡在本书编写和审阅过程中提出过意见和建议的同志，在此一并向他(她)们表示衷心的感谢。

本书在编写中得到了西北电讯工程学院教材科、图书馆的大力支持。有关领导部门、电子工业部教育局以及研究所和兄弟院校的很多领导、专家、教授和同志们对本书的编写曾给予了热情的鼓励，并提出过许多宝贵的意见。我们在此表示诚挚的感谢。

恳切希望读者继续对本书的缺点和错误予以指正。

编　者

1984年

目　录

第一章　计算机系统结构的基本概念

§1　计算机系统的多级层次结构 …… 1
§2　计算机系统结构、组成与实现 …… 5
2.1　计算机系统结构 …… 5
2.2　计算机组成与实现 …… 7
2.3　计算机系统结构、组成和实现三者的相互影响 …… 10
§3　软硬取舍、计算机系统与系统结构的设计方法 …… 13
3.1　软硬取舍的基本原则 …… 13
3.2　计算机系统的设计思路与系统结构设计的步骤 …… 15
3.2-1　计算机系统的设计思路 …… 15
3.2-2　计算机系统结构的设计步骤 …… 18
§4　软件、应用、器件对系统结构的影响 …… 19
4.1　系统结构设计中必须注意解决好软件的可移植性 …… 19
4.1-1　统一高级语言 …… 19
4.1-2　采用系列机思想 …… 20
4.1-3　模拟与仿真 …… 23
4.2　应用对系统结构的影响 …… 25
4.3　器件发展对系统结构的影响 …… 29
4.3-1　器件的功能和使用方法的变化会影响系统结构和组成 …… 29
4.3-2　器件的发展是推动系统结构和组成前进的关键因素 …… 31
4.3-3　器件的发展改变了逻辑设计的传统方法 …… 32
§5　系统结构和并行性的发展及计算机系统的分类 …… 33
5.1　Von Neumann 型计算机 …… 34
5.2　计算机系统结构中并行性的发展 …… 36
5.2-1　并行性概念 …… 36
5.2-2　计算机系统结构向并行处理系统的发展 …… 39
5.3　计算机系统的分类 …… 42
习题 …… 45
主要参考文献 …… 46

第二章　指令与寻址

§1　数据表示 …… 48
1.1　数据表示与数据结构 …… 48
1.2　浮点数尾数的基值大小和下溢处理方法选择 …… 52
1.2-1　浮点数尾数基值的选择 …… 53
1.2-2　浮点数尾数的下溢处理方法 …… 57
1.3　自定义数据表示与向量数据表示 …… 60
1.3-1　自定义数据表示 …… 60
1.3-2　向量数据表示 …… 63
§2　寻址方式 …… 65
2.1　寻址方式分析 …… 65
2.2　逻辑地址与主存物理地址 …… 68
§3　指令系统的设计和改进 …… 71
3.1　指令格式的优化 …… 71
3.1-1　哈夫曼压缩概念 …… 71
3.1-2　操作码的优化表示 …… 73
3.1-3　指令字格式的优化 …… 76
3.2　按增强指令功能的方向发展与改进指令系统 …… 79
3.2-1　面向目标程序的优化实现来改进 …… 80
3.2-2　面向高级语言的优化实现来改进 …… 84
3.2-3　面向操作系统的优化实现来改进 …… 90
3.3　按简化指令功能的方向发展和改进指令系统 …… 96

3.3-1 精简指令系统思想的提出 …… 96
3.3-2 RISC 结构采用的基本技术 …… 98
3.3-3 RISC 技术的发展 …… 101
习题 …… 105
主要参考文献 …… 107

第三章 输入输出系统

§1 引言 …… 109
§2 总线设计 …… 110
2.1 总线的类型 …… 111
2.2 总线的控制方式 …… 112
2.3 总线的通讯技术 …… 114
2.3-1 同步通讯 …… 114
2.3-2 异步通讯 …… 115
2.4 数据宽度与总线线数 …… 116
2.4-1 数据宽度 …… 116
2.4-2 总线的线数 …… 117
§3 中断系统 …… 118
3.1 中断的分类和分级 …… 118
3.2 中断系统的软硬功能分配 …… 123
§4 通道处理机 …… 124
4.1 工作原理 …… 124
4.2 通道流量的分析 …… 128
§5 外围处理机 …… 130
习题 …… 132
主要参考文献 …… 134

第四章 存贮体系

§1 引言 …… 136
1.1 存贮器容量、速度、价格的矛盾 …… 136
1.2 并行主存系统频宽的分析 …… 137
1.3 存贮体系的形成与发展 …… 141
1.4 存贮体系的性能参数 …… 143
§2 虚拟存贮器 …… 145
2.1 不同的虚拟存贮管理方式 …… 145
2.1-1 段式管理 …… 146
2.1-2 页式管理 …… 149
2.1-3 段页式管理 …… 150
2.2 页式虚拟存贮器构成 …… 152
2.2-1 地址的映象和变换 …… 152
2.2-2 替换算法 …… 156
2.2-3 虚拟存贮器工作的全过程 …… 162
2.3 页式虚拟存贮器实现中的问题 …… 163
2.3-1 页面失效的处理 …… 163
2.3-2 提高虚拟存贮器等效访问速度的措施 …… 164
2.3-3 影响主存命中率和 CPU 效率的某些因素 …… 170
§3 高速缓冲存贮器(Cache) …… 172
3.1 基本结构 …… 172
3.2 地址的映象与变换 …… 174
3.2-1 全相联映象和变换 …… 174
3.2-2 直接映象及其变换 …… 175
3.2-3 组相联映象及其变换 …… 176
3.2-4 段相联映象 …… 180
3.3 替换算法的实现 …… 180
3.3-1 堆栈法 …… 181
3.3-2 比较对法 …… 182
3.4 Cache 的透明性及性能分析 …… 183
3.4-1 Cache 的透明性分析 …… 183
3.4-2 Cache 的取算法 …… 186
3.4-3 任务切换对失效率的影响 …… 187
3.4-4 影响 Cache 存贮器性能的因素 …… 188
3.5 "Cache-主存-辅存"存贮层次 …… 190
§4 主存保护 …… 191
习题 …… 193
主要参考文献 …… 197

第五章 重叠、流水和向量处理机

§1 重叠解释方式 …… 199
1.1 基本思想和一次重叠 …… 199
1.2 相关处理 …… 202
1.2-1 指令相关的处理 …… 202
1.2-2 主存空间数相关的处理 …… 202
1.2-3 通用寄存器组相关的处理 …… 203
§2 流水方式 …… 207
2.1 基本概念 …… 207

2.1-1 流水是重叠的引伸 …… 207
2.1-2 流水线的分类 …… 208
2.1-3 阵列流水线 …… 213
2.2 主要性能及其分析 …… 215
2.2-1 吞吐率 …… 215
2.2-2 效率 …… 219
2.2-3 流水线工作举例 …… 220
2.3 相关处理和控制机构 …… 222
2.3-1 局部性相关的处理 …… 222
2.3-2 全局性相关的处理 …… 226
2.3-3 流水机器的中断处理 …… 229
2.3-4 流水线调度 …… 230
§3 向量的流水处理与向量处理机 …… 234
3.1 向量的流水处理 …… 234
3.2 向量处理机 …… 236
习题 …… 246
主要参考文献 …… 250

第六章 并行处理机和相联处理机

§1 并行处理机原理 …… 251
1.1 并行处理机的构形与特点 …… 251
1.1-1 并行处理机的基本构形 …… 251
1.1-2 并行处理机的特点 …… 253
1.2 并行处理机的算法 …… 254
1.2-1 ILLIAC Ⅳ的处理单元阵列结构 …… 254
1.2-2 阵列处理机的算法举例 …… 255
1.3 SIMD 计算机的互连网络 …… 259
1.3-1 互连网络的设计目标及互连函数的表示 …… 259
1.3-2 单级互连网络 …… 260
1.3-3 基本的循环互连网络和多级互连网络 …… 263
1.3-4 全排列网络 …… 269
1.4 并行存贮器的无冲突访问 …… 269
§2 并行处理机举例 …… 272
2.1 MPP 位平面阵列处理机 …… 272
2.2 DAP 阵列处理机 …… 274
2.3 BSP 科学处理机 …… 276
§3 相联处理机 …… 278
3.1 概述 …… 278
3.1-1 相联处理机的特点和组成 …… 278
3.1-2 相联存贮器的组成及相联处理机的结构类型 …… 279
3.2 相联检索算法 …… 281
3.3 相联处理机结构举例 …… 283
3.3-1 PEPE 系统 …… 283
3.3-2 STARAN 系统 …… 285
习题 …… 286
主要参考文献 …… 289

第七章 多处理机

§1 多处理机的特点及主要技术问题 …… 291
§2 多处理机的硬件结构 …… 292
2.1 紧耦合和松耦合 …… 292
2.1-1 紧耦合 …… 293
2.1-2 松耦合 …… 294
2.2 机间互连的形式 …… 296
2.2-1 总线形式 …… 296
2.2-2 交叉开关形式 …… 297
2.2-3 多端口存贮器形式 …… 298
2.2-4 开关枢纽结构形式 …… 299
§3 程序并行性 …… 300
3.1 并行算法的研究思路 …… 300
3.1-1 算术表达式的并行运算 …… 300
3.1-2 递归程序的并行性 …… 302
3.2 程序并行性的分析 …… 305
3.3 并行程序设计语言 …… 307
§4 多处理机的操作系统 …… 311
4.1 多处理机操作系统的类型 …… 312
4.1-1 主从型 …… 312
4.1-2 各自独立型 …… 312
4.1-3 浮动型 …… 313
4.2 资源分配和进程调度 …… 313
习题 …… 315
主要参考文献 …… 318

第八章 具有非诺依曼型结构特点的计算机

§1 基于面向对象程序设计的计算机 …… 320
1.1 面向对象的程序设计 …… 320
1.2 基于面向对象的程序语言的

计算机结构 …… 322
§2 基于数据驱动的数据流机 …… 324
2.1 数据驱动的概念 …… 324
2.2 数据流程序图和语言 …… 327
2.2-1 数据流程序图 …… 327
2.2-2 数据流语言 …… 330
2.3 数据流计算机的结构 …… 332
2.3-1 静态数据流机 …… 332
2.3-2 动态数据流机 …… 333
2.4 数据流机器存在的问题 …… 336
§3 基于面向函数程序设计的归约机 …… 337
3.1 函数式程序设计语言 …… 337
3.2 面向函数程序设计的归约机 …… 339
3.2-1 归约机的基本结构特点 …… 339
3.2-2 串归约机 …… 340
3.2-3 图归约机 …… 342
§4 基于面向智能信息处理的智能机 …… 343
4.1 智能信息处理与智能机 …… 343
4.2 智能机的结构及所用的机器语言 …… 344
4.2-1 智能机的结构 …… 344
4.2-2 逻辑程序设计语言 …… 346
4.2-3 智能计算机的进展 …… 347
习题 …… 348
主要参考文献 …… 349

第一章　计算机系统结构的基本概念

本章首先提出可以把计算机系统看成是按功能划分的多级层次结构，由此引出计算机系统结构的定义，并说明了结构、组成、实现三者的含义和关系，以及计算机软、硬件功能分配的一般原则；然后叙述软件、应用、器件与计算机设计对系统结构的影响；最后介绍计算机系统中并行性的发展以及计算机系统的分类。目的是在学习后面各章之前，能对计算机系统结构有一个基本的了解。

§1　计算机系统的多级层次结构

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

随着计算机系统的发展，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、应用语言这样一个由低级向高级发展的过程，后者均以前者为基础，又比前者功能更强，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级，最低层的语言功能最简单。对使用某一层(级)语言编程的程序员来讲，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，就好像他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。

实际上，只有二进制机器指令(即通常简称的机器语言)是与机器硬件直接对应，并被其直接识别和执行的，然而使用机器语言编程既不方便，也无法适应解题需要和计算机应用范围的扩大。汇编语言是一种符号式程序设计语言，给程序员编程提供了方便，尽管其每个语句仍基本上与机器指令对应，却并不能被机器硬件直接识别和执行。那么，为什么汇编语言源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？我们可以把这想像成是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”的机器，如图 1.1 所示。于是从功能上，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，不用了解实际机器级 M1。在计算机系统上运行汇编语言(L2)源程序是先整个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此目标程序来获得结果。这样一种把源程序先变换成目标程序，而后再在机器上执行目标程序以获得结果的技术称为翻译(Translation)。

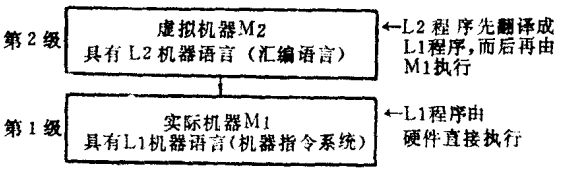

图 1.1　汇编虚拟机器的实现

由于汇编语言的语法、语义结构仍然和二进制机器语言的基本相同，与解题所需的差别较大，于是又进一步出现了面向题目和过程的高级语言，如 BASIC、FORTRAN、

PASCAL 等。同样，对高级语言程序员来说就好像他有了一台使用高级语言作为其机器语言的虚拟机器，可以把它想像成在汇编语言虚拟机器级之上又出现了高级语言虚拟机器级。高级语言源程序的实现是先经编译程序整个地翻译成汇编语言程序或是机器语言程序，再逐级或越级向下实现，如图 1.2 所示。

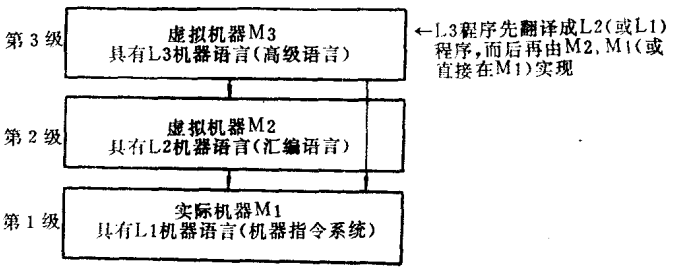

图 1.2　高级语言虚拟机器的实现

这种层次概念还可引伸到机器内部。对于采用微程序控制的机器，每条机器指令对应于一串微指令(一段微程序)，每条微指令执行一些如控制各种数据传送通路之类的最基本操作。这样，又可以把图 1.2 中的实际机器级分解成如图 1.3 所示的传统机器级 M1 和微程序机器级 M0 构成的二级层次结构。微程序机器级的机器语言是微指令系统，只是机器语言程序的实现并不是像汇编语言程序或高级语言程序那样先翻译成低一级语言的等效程序后再执行，而是当执行到某条机器指令时控制转入执行相应的一串微指令，实现完这条机器指令后，再由程序内的下条机器指令控制转入实现它的另一串微指令。这种实现过程称为解释(Interpretation)。如

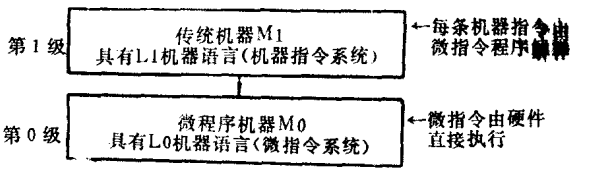

图 1.3　机器指令由微指令程序解释

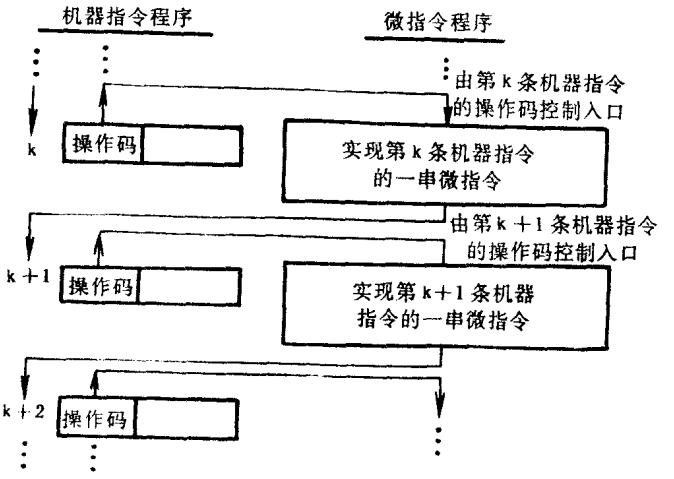

图 1.4　用微指令程序解释机器指令

图 1.4 所示，是一种边解释边执行的方式，在这个过程中不产生翻译出来的中间程序。这个过程也可理解为在 M_1 机器（这里是微程序机器 M0）上，用一串 M_1 机器指令（这里是微指令）的执行来仿真比它高级的机器的一条语句（这里是比 M0 高一级的 M1 机器指令）。应当指出的是，不用微程序控制的机器没有微程序机器级，传统机器的指令直接用硬件实现而不用解释程序解释，另一方面有些高级语言也可用解释方法实现，不一定非用翻译技术。

翻译和解释是语言实现的两种基本技术。解释比翻译费时，但省存贮空间。对于微程序控制的机器，在高级语言的实现过程中，通常这两种技术都要用到。即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。同时，由高级语言程序变换成机器语言程序的过程中也不都是采用翻译技术，往往也是翻译和解释两种技术并用。一般是先把高级语言源程序翻译成易于执行的某种中间形式，而后再用机器语言程序解释它。至于机器语言程序的每条机器指令，则又经微程序解释。

上面分析了可以把计算机系统看成由高级语言虚拟机器、汇编语言虚拟机器、传统机器语言机器和微程序语言机器所构成的多级层次结构。那么，操作系统应处在这个层次结构中的什么位置呢？

虽然操作系统具有提高计算机系统的效率以及控制汇编语言、高级语言等的实现和作业的运行等功能，从这点看，似乎可以把它置于前述按语言结构划分的层次结构之外，作用于所有各级；虽然目前很多机器的操作系统已经不用汇编语言而改用高级语言（不是用面向解题的算法语言，而是用面向系统软件的高级语言，如 C 语言）来编写，从这点看，似乎又可以把它置于高级语言机器级之上，但从实质上来看，操作系统是传统机器的引伸，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，如文件结构与文件管理的基本操作、存贮体系以及多道程序和多重处理所用的某些操作等等，这些可以看成是操作系统的指令系统，其中包含作业控制语言等的操作和数据结构，它们在许多机器上是经机器语言程序解释实现的。因此，操作系统级放在传统机器级之上，汇编语言机器级之下更为合适。当然，有些机器语言指令，如某些具体的 I/O 操作指令是要被操作系统“挡”住的，但大部分机器语言指令，如运算类指令等等，却是原样穿过操作系统级，而且也包括在操作系统级的指令系统之内。

在高级语言机器级之上还可以有应用语言虚拟机器，它是为使计算机满足某种应用而专门设计的，如设计成专门用于管理、人工智能、图像处理、辅助设计等。这种虚拟机所用的语言是面向某种应用环境的应用语言。用应用语言编写的程序一般经应用程序包翻译成高级语言程序后，再逐级向下地实现。

总之，一个现代的计算机系统可以从功能上看成是如图 1.5 所示形式的多级层次结构。还可以考虑在第 0 级之下再分级。例如，有的微程序机器采用二级微程序（称为微程序与毫微程序）去解释机器指令。每条微指令不是直接由硬件执行，而是由比它低一级的毫微程序解释。图中每级对应一类机器，各有其自己的机器语言。在这里，“机器”被定义为能存贮和执行程序的算法和数据结构的集合体。各级机器的算法和数据结构的实现方法不同，目前来看，M0 是硬件实现，M1 是微程序（固件）实现，M2 到 M5 大多是软件实现的。我们称由软件实现的机器为虚拟机器，以区别于由硬件或固件实现的实际机器。要

注意，机器的实现和题目的得到解答不是一回事，后者是要从你用的那级开始逐级变换，直至到达最低级，它需经硬件实现才能得到；而前者主要表现为如何把该级的程序或是翻译成比它低级的语言的程序，或是由低级的程序所解释。虚拟机器也不一定全都由软件实现，有些操作也可能是用固件或硬件实现，如操作系统中的某些命令可以由比它低二级的微程序解释。甚至可以设想直接用微程序或硬件来实现高级语言机器，直接用固件来实现操作系统机器。

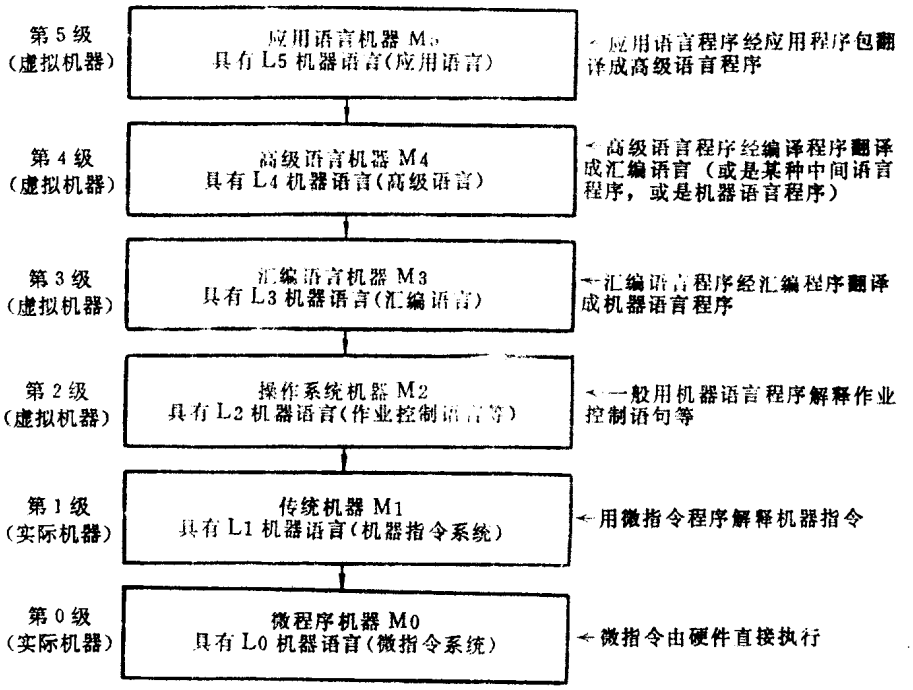

图 1.5　计算机系统的多级层次结构

采用何种实现方式，要从整个计算机系统的效率、速度、造价、资源的使用状况等方面全面考虑，而且要软件、硬件(包括固件)综合平衡。实际上，软件和硬件在逻辑功能上是等效的。在原理上，软件实现的功能可以用硬件或固件完成，硬件实现的功能也可以由软件的模拟来完成，只是其性能、价格、实现的难易程度等不同而已。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，如图 1.6 所示。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，有时软硬件界面还是模糊不清的，一个功能很难说哪些是硬件完成，哪些是软件完成的。例如，中断处理、存贮管理等许多功能就是这样的。选择什么样的软、硬分配比例，主要取决于在现有硬、器件状况下的性能价格比。应当在满足应

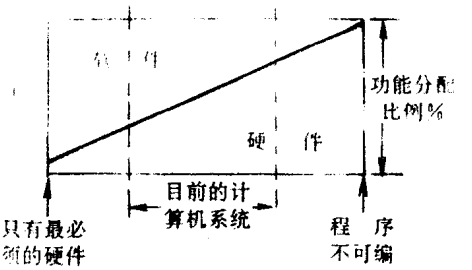

图 1.6　计算机系统的软、硬件功能分配

用的前提下，充分利用硬、器件技术的进展，使系统达到较高的性能价格比。此外，对计算机系统使用者来说，还要考虑他所直接接触到的那级机器(或是应用语言机器级，或是高级语言机器级)的状况。

从概念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有利于正确理解软件、硬件、固件在计算机系统中的地位和作用。可以看出，微程序机器级、传统机器级、操作系统机器级不是直接为应用程序员解题设计的，主要是为运行支持更高层次机器级程序所必须的解释程序和翻译程序而设计的，以便能设计和实现新的虚拟机器级。然而，汇编语言机器级、高级语言机器级、应用语言机器级则主要是为应用程序设计人员求解问题设计的。这也有助于理解各种语言的实质和实现途径。计算机语言已越来越接近于人的自然语言以便于人们使用，然而却总是通过翻译成低级的语言，或由低级语言解释来实现。为了便于翻译和解释，相邻级语言的语义差距就不能太大。

把计算机系统看成是多级层次结构还推动了计算机系统结构的发展。例如，可以为应用语言级、高级语言级、操作系统级提供更多更好的硬件支持，或就用硬件或固件来实现，发展了所谓高级语言机器、操作系统计算机结构。既然层次结构中的每一级都有它自己的用户，自己的实现方法，自己的指令系统，因此就可以设想让各级虚拟机器用真正存在的处理机来代替，摆脱以往各级功能都在同一台实际机器上实现的状况，发展了多处理机系统、分布处理系统、计算机网等系统结构。可以在一台真正的宿主机上通过模拟或仿真另一台不同的假想机器，来推动自虚拟机技术，多种操作系统共同执行等技术的采用，从而促进软件移植、计算机系统性能评价、计算机设计自动化等方面的发展。

这种多级层次结构的概念也有助于我们理解计算机系统结构的定义，更好地进行计算机系统的设计。那么，怎样从层次结构来定义"计算机系统结构"呢？

§2 计算机系统结构、组成与实现

2.1 计算机系统结构

计算机系统结构(Computer Architecture)也称计算机体系结构。这个名词从70年代开始已被广泛采用，但由于器件技术的迅速发展，计算机软、硬件界面在动态变化，使得不少人对计算机系统结构的含义理解不尽一致。

1964年G. M. Amdahl在介绍IBM 360系统时提出：计算机系统结构是从程序员所看到的计算机的属性，即程序员编写出的能在机器上正确运行的程序所必须了解到的概念性结构和功能特性。然而现在从计算机系统的层次结构概念出发，不同级的程序员所看到的计算机属性显然是不一样的。

如图1.7所示，NOVA机和PDP-11机，DJS-1000机和DJS-2000机是不同的计算机，它们的指令系统(如操作类型和寻址方式)、寄存器组织、输入／输出设备连接方式都是很不相同的。就是说，从机器语言程序员或汇编语言程序员看，它们的属性是很不相同的。然而，从使用高级语言(如FORTRAN)的程序员来看，这二种机器的属性却几乎没有差别，只要都遵守FORTRAN语言的规定编写程序就可以了。说几乎没有差别，是指不同厂家生产的机器上同一种高级语言使用了"方言"而难以完全通用，但这种差别是很细微的。这反映出传统机器级存在的差别是高级语言程序员所不需要看见的，对他是透

明的。本来存在的事物或属性，从某个角度去看却好像不存在，我们称这种概念为透明性(Transparency)概念。一般，低层机器的属性对高层机器程序员基本上是透明的。如传统机器级的概念性结构和功能特性对汇编语言程序员来说大部分是要看到的，但对高级语言程序员来说却基本上是透明的。

这样，从不同层次(级)的程序员看，计算机的属性是不同的。这个属性就是计算机系统不同层次的界面。所谓"系统结构"就是指的计算机系统中对各级之间界面的定义及其上下的功能分配。所以，各级都有它自己的系统结构。系统结构设计者要研究对某级哪些应当透明，哪些不应当透明。透明的好处是可以不用管它，简化了该级的设计，反过来由于看不到，无法控制会带来不利。因此，正确合理地进行透明性的分析、取舍是非常重要的。

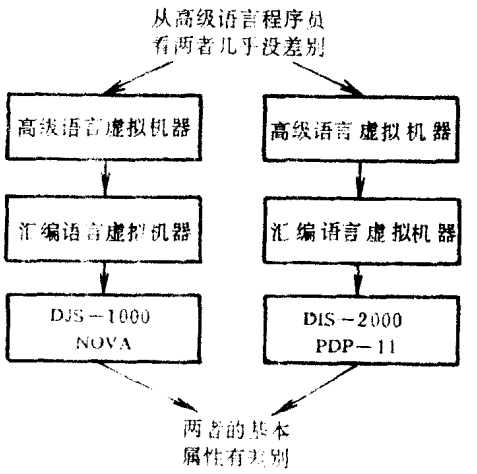

图 1.7　从高级语言程序员看的计算机属性

当时，Amdahl 列举出程序员所看到的属性有：数据是如何表示(即数据表示)、如何被访问的(即寻址方式)，能对这些数据进行什么样的运算和如何控制这些运算的执行(即指令系统)等等。可以看出，他所列举的是现在的传统机器语言程序员为使其所编写的程序能在机器上正确运行所看到的属性。这些属性不只是与实际执行指令的处理机有关，还包括运行机器语言程序要看到的处理机、存贮系统、I／O 联结方式和中断机构等。

本书所讲的计算机系统结构，指的就是图 1.5 中传统机器级(上面简称机器级)的系统结构。其界面之上的功能指的是所有软件的功能，即包括操作系统级、汇编语言级、高级语言级和应用语言级的所有功能。界面之下的功能指的是所有硬件和固件的功能。因此，这个界面实际上是软件与硬件／固件的交界面，如图 1.8 所示。所以，计算机系统结构研究的就是软、硬件功能分配以及对机器级界面的确定，即由机器语言设计者或编译程序设计者所看到的机器物理系统的抽象或定义。它是机器语言程序设计者或是编译程序生成系统为使其所设计或生成的程序能在机器上正确运行，所需看到和遵循的计算机属性。它不包括机器内部的数据流和控制流、逻辑设计和器件设计等，这些属于我们稍后要讲的计算机组成和实现。

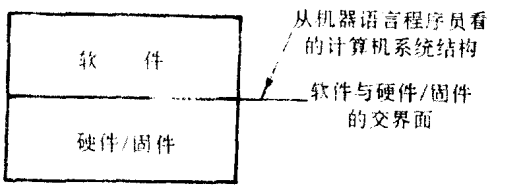

图 1.8　计算机系统结构是软、硬／固件的交界面

对于目前的通用型机器，计算机系统结构一般包括：

数据表示(硬件能直接识别和处理的数据类型和格式等)；

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)；

寄存器组织(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）；

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）；

存贮系统（包括最小编址单位、编址方式、主存容量、最大可编址空间等）；

中断机构（中断的类型、中断分级、中断处理程序的功能和入口地址等）；

机器工作状态（如管态、目态等）的定义和切换；

机器级的I／O结构（包括I／O的联结方式、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）；

信息保护（包括保护方式、硬件对信息保护的支持等）；等等。

据此，DJS-1000计算机与DJS-2000计算机，NOVA机与PDP-11机显然有不同的机器级界面，即有不同的计算机系统结构。为了叙述简便，如果没有专门说明，书中的“系统结构”实际上指的就是“计算机系统结构”。

2.2 计算机组成与实现

计算机组成（Computer Organization）指的是计算机系统结构的逻辑实现，包括机器级内的数据流和控制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（Computer Implementation）则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。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。

下面我们举例说明什么是计算机系统结构、计算机组成和计算机实现。

例如，指令系统的确定属计算机系统结构，指令的实现，如取指令、取操作数、运算、送结果等的具体操作及其排序方式属计算机组成，而实现这些指令功能的具体电路、器件的设计及装配技术等则属计算机实现。

又如，确定是否要有乘法指令属计算机系统结构，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属计算机组成，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、数量和使用何种价格的器件等的确定）及所用微组装技术等则属计算机实现。

再如，对主存系统，主存容量与编址方式（按位、按字节还是按字访问等）的确定属计算机系统结构，为要达到所定的性能价格比，主存速度应多快，在逻辑结构上需采用什么措施（如多体交叉存贮等）属计算机组成，而主存系统的物理实现，如存贮器器件的选定、逻辑电路的设计、微组装技术的选定则属计算机实现。

可以看出，具有相同结构（如指令系统相同）的计算机可以因为速度要求不同等因素而采用不同的组成。例如，指令的取出、译码、取操作数、运算、存结果可以顺序进行，也可以让它们在时间上重叠进行以提高执行速度。又如乘法指令采用专用乘法器实现，也可以采用经加法器重复相加、右移实现，这取决于要求的速度、程序中乘法指令的出现频度及所采用的乘法运算方法。速度要求高、出现频度高的可用专用乘法器，如果出现频度很低，采用后种方法速度的下降并不明显，却可以显著降低价格。

同样，一种计算机组成可以采用多种不同的计算机实现。例如，主存器件可以用双极

型的，也可以使用MOS型的，可以采用大规模集成电路单片，也可以用小规模集成电路多片组搭。显然这取决所要求达到的性能价格比和器件技术的现状。

就计算机组成设计而言，要解决的问题是在所希望达到的性能价格比下，怎样最佳、最合理地把各种设备和部件组成计算机，以实现所确定的系统结构。近40年里，计算机组成设计主要围绕如何提高速度，着重从提高操作的并行度、重叠度，以及功能的分散和专用功能部件来设计。

一般，计算机组成设计要确定的方面包括：

数据通路宽度(数据总线上一次并行传送的信息位数)；

专用部件的设置(设置哪些专用部件，如乘除法专用部件、浮点运算部件、字符处理部件、地址运算部件等，每种专用部件个数等等，这些都取决于所需达到的机器速度、专用部件的使用频度及允许的价格等)；

各种操作对部件的共享程度(共享程度高，尽管这些操作在逻辑上互不相关，也只能分时使用，限制了速度。可用设置多个部件降低共享程度，用提高操作并行度来提高速度，但价格也将提高)；

功能部件的并行度(功能部件的控制和处理方式是采用顺序串行，还是采用重叠、流水、分布处理)；

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)；

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顺序，这可以是随机、先进先出、先进后出、优先级、循环队等不同方式)；

预估、预判技术(采用什么原则来预测未来的行为，以优化性能和优化处理)；

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)；

等等。

这里，对推动计算机组成技术发生重大进展的微程序技术稍许多讲几句。

前已讲过(参看图1.3)，采用微程序控制的机器是在微程序机器M0上，用一串M0级机器指令(即微指令)的执行来仿真传统机器级M1的一条机器指令。我们称M0级机器为微程序宿主机。采用微程序组成方式的机器实质上是在某个宿主机上，用仿真方法实现计算机系统结构的指令系统。

如果说，计算机头20年的改进主要是为了提高速度的话，那么微程序技术的提出和采用最初却主要是为了使控制器规整，以便于设计和生产。如图1.9所示，它把存贮器技术引入到控制器内(有人因此称之为存贮逻辑)，把硬联控制器网络的复杂逻辑联结，转变为控制存贮器中存贮的复杂码点(各种微指令码点)，从而用规整的、可大量生产的存贮器件来取代杂乱的硬联控制网络，而且，微指令寄存器

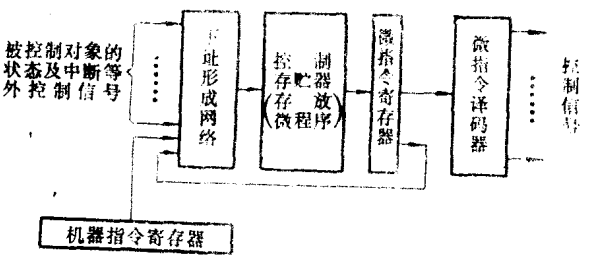

图1.9 微程序控制器