



计算机系统结构

(第2版)

陈智勇 编著



卓越工程师培养计划『十二五』规划教材

創工
新程

Engineering Innovation



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

卓越工程师培养计划“十二五”规划教材

计算机系统结构

(第2版)

陈智勇 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书系统地介绍了计算机系统结构的基本概念、设计原理和分析方法,以及有关的先进技术和最新的发展趋向。

全书共分9章。第1章介绍计算机系统层次结构,计算机系统结构、组成、实现的定义及相互关系,计算机性能,以及计算机系统结构的分类。第2章介绍指令格式的优化,以及CISC、RISC、退耦CISC/RISC、后RISC的有关概念及其采用的相关技术。第3章介绍存储体系,页式虚拟存储器和Cache存储器。第4章介绍标量流水线技术,超标量超流水技术,VLIW体系结构,以及先进的流水线调度技术。第5章介绍并行处理技术,SIMD并行处理机及互连网络。第6章介绍向量的流水处理,以及提高向量处理性能的常用技术。第7章介绍多处理机结构,多处理机Cache的一致性,多处理机性能模型,以及并行计算机模型。第8章主要介绍并行算法与并行编程的基础知识。第9章主要介绍并行计算机系统结构的新发展。

本书内容丰富、取材先进,在阐述基本原理的基础上,力图给出设计方法和实例,以帮助读者更好地理解一些比较抽象的概念。每章均有一定数量的例题和习题,书后附有习题的参考答案。本书免费提供配套电子课件,请登录华信教育资源网下载(www.hxedu.com.cn)。本书可作为高等学校计算机专业本科教材,也可作为计算机相关专业研究生教材或科技人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

计算机系统结构 / 陈智勇编著. —2版. —北京:电子工业出版社, 2012.1

卓越工程师培养计划“十二五”规划教材

ISBN 978-7-121-15276-4

I. ①计… II. ①陈… 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242760号

责任编辑:冉 哲

特约编辑:冉 哲

印 刷:北京市顺义兴华印刷厂

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

开 本:787×1092 1/16 印张:18.5 字数:522千字

印 次:2012年1月第1次印刷

印 数:3000册 定价:33.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

前 言

实际应用对计算机系统高性能的要求、器件技术的迅速发展以及计算机软件技术的发展,加快了计算机系统结构的发展进程。新的系统结构技术不断涌现,并且很快被应用到新的计算机产品中。计算机系统结构的发展如此迅速,使得教材的编写很难反映计算机系统结构当前最新的发展状况。本书内容丰富、取材先进,在阐述基本原理的基础上,力图给出设计方法和实例,以帮助读者更好地理解一些比较抽象的概念。每章均有一定数量的例题和习题,书后附有习题的参考答案。

本书第一版于 2004 年由西安电子科技大学出版社出版,并被列为“面向 21 世纪高等学校计算机类专业系列教材”。该书已被许多学校选为教材,并于 2006 年获得省级优秀教材二等奖。为了跟踪和反映计算机技术的发展,结合本科生教学内容要求,以及同等学力人员申请硕士学位计算机科学与技术学科综合水平全国统一考试大纲和近几年的统考试题,在本书第一版的基础上主要进行了如下修改:

将第 2 章的退耦 CISC/RISC 体系结构由原来的 Pentium Pro 改为 Core;在第 3 章中补充了虚拟地址到 Cache 地址的两种转换方法;在第 4 章中删除了重叠解释方式,补充了非线性流水线的优化调度方法、转移预测技术、超标量流水线的调度等内容;在第 7 章中修改了多处理机系统的分类方法,补充了各种多处理机和多计算机系统的存储器体系结构,以及并行计算机模型;在第 8 章中增加了硬件并行性与软件并行性的分析;将第 9 章的内容替换成了并行计算机系统结构新发展概述,内容包括多线程技术、单芯片多处理器技术、机群系统、网格计算和云计算等。

本书在编写上具有如下 5 个特色:第一,取材先进,在“流水线计算机设计技术”、“并行处理技术”、“多处理机”、“并行算法与并行编程基础”、“并行计算机系统结构新发展概述”等章节都引用了近几年国内外计算机系统结构方面比较成熟的研究成果和先进技术;第二,有较宽的课程教学适应面,教师可根据学生的学历层次和教学大纲要求选择不同的章节教学,同时本教材还可作为计算机系统设计、并行算法、并程序序设计等课程的教学参考书;第三,例题丰富,解题过程详细,思路清晰,有助于对基本理论、基本方法的理解;第四,各章后的习题均附有参考答案,有助于读者自学。第五,本书免费提供配套电子课件,可登录华信教育资源网下载(www.hxedu.com.cn)。

本课程应在“数字逻辑”、“计算机组成原理”、“汇编语言程序设计”、“高级语言程序设计”、“数据结构”等课程之后开设,可在“操作系统”、“编译原理”课程之后,或与它们同时开设。本书可作为高等学校计算机专业本科生的教材以及相关专业的研究生教材。本课程的参考教学时间为 60 学时,可根据实际情况选择部分内容学习。

本书的编写和出版得到了桂林电子科技大学古天龙副校长的推荐和支持,电子工业出版社的编辑为本书的出版也做了大量的工作,在此表示衷心的感谢。

由于计算机技术的飞速发展,新的理论和技术层出不穷,教材难以反映系统结构的最新发展变化,再加上编者的水平有限,在教材中难免会存在不少问题和错误,恳请广大读者批评指正。

作者

目 录

第 1 章 基础知识	1
1.1 计算机系统层次结构	1
1.1.1 按功能划分的多级层次结构	1
1.1.2 按功能划分层次的好处	3
1.2 计算机系统结构	3
1.3 计算机组成	4
1.4 计算机实现	5
1.5 计算机性能	7
1.5.1 性能因子 CPI	7
1.5.2 计算机性能常用指标 MIPS 和 MFLOPS	7
1.5.3 计算机系统结构的性能评价标准	12
1.5.4 计算机系统设计的定量原理	14
1.6 计算机系统结构的分类	16
1.6.1 Flynn 分类法	16
1.6.2 冯氏分类法	17
习题 1	18
第 2 章 指令系统	20
2.1 指令格式的优化	20
2.1.1 操作码的优化表示	20
2.1.2 指令字格式的优化	23
2.2 CISC	25
2.2.1 CISC 的发展	26
2.2.2 CISC 的特点	26
2.3 RISC	27
2.3.1 RISC 结构采用的基本技术	27
2.3.2 经典 CISC 和纯 RISC 处理器体系结构的比较	30
2.3.3 退耦 CISC/RISC 体系结构	31
2.4 后 RISC	35
2.4.1 后 RISC 特征	36
2.4.2 几种后 RISC 机器的比较	36
习题 2	37
第 3 章 存储系统设计	39
3.1 存储系统原理	39
3.1.1 基本概念	39
3.1.2 多级存储层次	39
3.1.3 存储系统的性能参数	40

3.2 交叉访问存储器	42
3.2.1 主存系统的类型	42
3.2.2 高位交叉访问存储器	43
3.2.3 低位交叉访问存储器	43
3.2.4 拓宽存储器频宽的方法	44
3.3 页式虚拟存储器	45
3.3.1 虚拟存储器的工作原理	45
3.3.2 虚拟存储器的地址映像与变换	46
3.3.3 页面替换算法及其实现	49
3.3.4 提高虚拟存储器等效访问速度的措施	52
3.3.5 影响主存命中率的某些因素	56
3.4 Cache 存储器	58
3.4.1 Cache 存储器的工作原理	58
3.4.2 Cache 存储器的地址映像与变换	59
3.4.3 Cache 替换算法及其实现	65
3.4.4 Cache 的透明性分析	67
3.4.5 Cache 的取算法	68
3.4.6 影响 Cache 存储器性能的因素	69
3.4.7 物理 Cache 与虚拟 Cache	71
习题 3	72
第 4 章 流水线计算机设计技术	76
4.1 流水线的工作原理、特点及分类	76
4.1.1 流水线的工作原理	76
4.1.2 流水线的特点	77
4.1.3 流水线的分类	78
4.2 相关问题及解决方法	80
4.2.1 资源相关	81
4.2.2 数据相关	81
4.2.3 控制相关	83
4.3 线性流水线性能分析	83
4.3.1 吞吐率和加速比	83
4.3.2 效率	86
4.4 线性流水线性能分析举例	87
4.5 非线性流水线的调度技术	89
4.5.1 非线性流水线的基本概念	89
4.5.2 无冲突调度方法	90
4.5.3 优化调度方法	92
4.6 先进的流水技术:动态调度	94
4.6.1 流水的集中式动态调度:记分板机制	95
4.6.2 流水的分布式动态调度:Tomasulo 方法	96
4.7 转移预测技术	98

4.7.1	转移的影响	98
4.7.2	转移预测技术	98
4.7.3	转移目标缓冲器	99
4.8	超标量处理机和 VLIW 体系结构	100
4.8.1	超标量处理机	100
4.8.2	VLIW 体系结构	104
4.9	超流水线处理机	106
4.10	超标量超流水线处理机	106
习题 4		110
第 5 章	并行处理技术	113
5.1	计算机系统结构中并行性的发展	113
5.1.1	并行性的基本概念	113
5.1.2	实现并行性技术的途径	114
5.1.3	计算机系统结构中并行性的发展	114
5.2	SIMD 并行处理机	115
5.2.1	SIMD 并行处理机的基本结构与特点	115
5.2.2	阵列处理机 ILLIAC IV 的处理单元阵列结构	117
5.2.3	阵列处理机的并行算法	118
5.3	SIMD 计算机的互连网络	122
5.3.1	互连网络的设计准则	122
5.3.2	互连函数的表示	122
5.3.3	单级互连网络	123
5.4	网络特性	125
5.4.1	结点度和网络直径	125
5.4.2	聚集带宽和等分带宽	125
5.4.3	数据寻径功能	125
5.5	静态网络	126
5.5.1	线性阵列	127
5.5.2	环和带弦环	127
5.5.3	循环移数网络 and 全连接	128
5.5.4	树形和星形	128
5.5.5	胖树形	128
5.5.6	网格形和环网形	128
5.5.7	超立方体	129
5.5.8	带环立方体	129
5.5.9	k 元 n -立方体网络	129
5.6	动态网络	132
5.6.1	总线互连方式	132
5.6.2	交叉开关互连方式	134
5.6.3	多级网络互连方式	135
5.6.4	蝶式网络	142

5.6.5 组合网络	142
习题 5	144
第 6 章 向量处理机	148
6.1 向量处理的基本概念	148
6.1.1 向量流水处理的主要特点	148
6.1.2 向量处理机的基本系统结构	148
6.1.3 向量的处理方式	149
6.2 一般的向量处理机	151
6.2.1 寄存器组	151
6.2.2 多个单功能流水部件	153
6.2.3 运算流水线	153
6.3 提高向量处理性能的常用技术	154
6.3.1 多功能部件的并行操作	154
6.3.2 链接技术	155
6.3.3 条件语句和稀疏矩阵的加速处理方法	157
6.3.4 向量归约操作的加速方法	158
6.4 向量处理的性能评价	159
6.4.1 向量流水处理的时间	159
6.4.2 与向量长度有关的向量流水处理机性能参数	160
6.5 多向量多处理机	162
6.5.1 Cray Y-MP	162
6.5.2 C-90 和机群	163
6.5.3 VP2000	164
6.5.4 VPP500	164
习题 6	165
第 7 章 多处理机	168
7.1 多处理机结构	168
7.1.1 多处理机的特点	168
7.1.2 多处理机的硬件结构	169
7.1.3 UMA、NUMA 和 NORMA 模型	173
7.2 多处理机的 Cache 一致性	174
7.2.1 Cache 一致性问题的原因	174
7.2.2 监听一致性协议	176
7.2.3 MESI 监听协议	177
7.2.4 基于目录的协议	180
7.3 程序的划分和调度	183
7.3.1 粒度的组合和调度	183
7.3.2 静态多处理机调度	184
7.4 多处理机性能模型	187
7.4.1 基本模型	187
7.4.2 N 台处理机系统的基本模型	188

7.4.3	随机模型	189
7.4.4	通信开销为线性函数的模型	191
7.4.5	一个完全重叠通信的理想模型	191
7.4.6	一个具有多条通信链的模型	193
7.5	并行计算机模型	194
7.5.1	语义属性	194
7.5.2	抽象机器模型	195
7.5.3	物理机器模型	198
	习题 7	201
第 8 章	并行算法与并行编程基础	203
8.1	并行算法的基础知识	203
8.1.1	并行算法的定义和分类	203
8.1.2	进程中的同构性	204
8.1.3	并行算法的表达	204
8.1.4	并行算法中的同步与通信	205
8.2	同步技术	206
8.2.1	原子性	207
8.2.2	控制同步	207
8.2.3	数据同步	208
8.2.4	高级同步结构	209
8.2.5	低级同步原语	211
8.3	并行性条件	216
8.3.1	程序的并行性分析	216
8.3.2	硬件并行性与软件并行性	219
8.4	并行编程	220
8.4.1	并行编程概述	221
8.4.2	并行编程方法	221
8.4.3	并行算法范例	223
8.5	并行编程模型	225
8.5.1	隐式并行性	225
8.5.2	显式并行模型	225
	习题 8	230
第 9 章	并行计算机系统结构新发展概述	232
9.1	多线程技术	232
9.1.1	多线程处理器	232
9.1.2	同时多线程处理器	233
9.1.3	多线程处理器芯片实例	235
9.2	单芯片多处理器技术	237
9.2.1	单芯片多处理器	237
9.2.2	单芯片多处理器芯片实例	240
9.3	机群系统	243

9.3.1 机群体系结构	244
9.3.2 机群的分类	245
9.3.3 机群设计要考虑的问题	245
9.3.4 对机群可用性的支持	246
9.4 网格计算	248
9.4.1 网格计算的定义	248
9.4.2 网格计算的特点	249
9.4.3 网格计算的体系结构	249
9.4.4 Globus Toolkit	251
9.4.5 网格计算系统的功能	252
9.4.6 网格计算的关键技术	253
9.5 云计算	253
9.5.1 云计算的定义	253
9.5.2 云计算与网格计算的比较	254
9.5.3 Google 的云计算平台	255
习题 9	257
附录 A 习题参考答案	258
参考文献	286

第1章 基础知识

本章首先从多级层次结构的角度定义计算机系统结构,并分析计算机系统结构、计算机组成与计算机实现的含义、研究的内容以及三者之间的相互关系。然后,介绍计算机的性能指标和性能评价标准,并讨论计算机系统设计的定量原理。最后,介绍计算机系统结构的两种分类方法。

1.1 计算机系统层次结构

1.1.1 按功能划分的多级层次结构

现代计算机是通过执行指令来解决问题的,它由软件和硬件两大部分组成。描述一个任务如何实现的指令序列称为程序。所有程序在执行前都必须转换成计算机能识别且能直接执行的机器指令。这些机器指令与机器硬件直接对应,并能被其直接识别和执行。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。这个问题可从两方面去解决,前提是都需要设计一个比机器指令更便于使用或编程的指令集合,由它构成新的语言,例如汇编语言。汇编语言是一种符号语言,给程序员编程提供了方便,尽管每条语句仍基本上与机器指令相对应,却并不能被机器直接识别和执行。用汇编语言开发的程序需要进行某种转换才能在实际机器上执行:一种是翻译(translation),即在执行汇编语言源程序之前生成一个等价的机器语言指令序列来替换它,生成的程序全部由机器指令组成,计算机执行等效的机器语言源程序,而不是原来的汇编语言源程序,也即把源程序先转换成目标程序,而后再在机器上执行目标程序以获得结果;另一种是解释(interpretation),即用机器指令写一个程序,将汇编语言源程序作为输入数据,按顺序检查它的每条指令,然后直接执行等效的机器指令序列来解决问题。

汇编语言源程序可以在机器上运行并获得结果,是因为有汇编程序的支持。在汇编语言程序设计者看来,就好像有了一台用汇编语言作为机器语言的机器。这里的机器是指能存储和执行程序的算法和数据结构的集合体。我们把以软件为主实现的机器称为虚拟机器,而把由硬件和固件实现的机器称为实际机器。显然,虚拟机器的实现是构筑在实际机器之上的。图 1.1 给出了虚拟机器与实际机器构成的一个简单的层次结构。

语言与虚拟机之间存在着重要的对应关系,每种机器都拥有由它能执行的指令组成的机器语言。同时,语言也定义了机器,即机器要能执行这种语言所写的程序。有 n 层不同的语言,就对应有 n 层不同的虚拟机。在图 1.2 中给出了典型的现代计算机系统的多级层次结构。例如,汇编语言(L3)是面向机器的一种符号语言,其语法、语义结构仍

然和二进制机器语言的基本相同,但与解题所需的差别较大,于是进一步出现了面向题目和过程的高级语言(L4);在高级语言的基础上还出现了面向应用的应用语言(L5),因此可以得到多个层次的虚拟机器;而对于实际机器级,若采用微程序(L0)控制,它又可分解成传统机器级 M1 和微程序级 M0;虽然目前很多机器上的操作系统(L2)已不再用汇编语言编写,而是用面向系统软件的高级语言,如 C 语言编写,但从实质上看,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

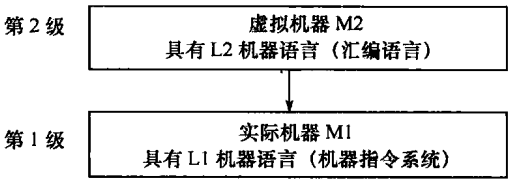


图 1.1 虚拟机器和实际机器层次结构举例

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,如文件管理、进程管理、中断管理、作业控制、存储管理和输入/输出等,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,汇编语言级之下。

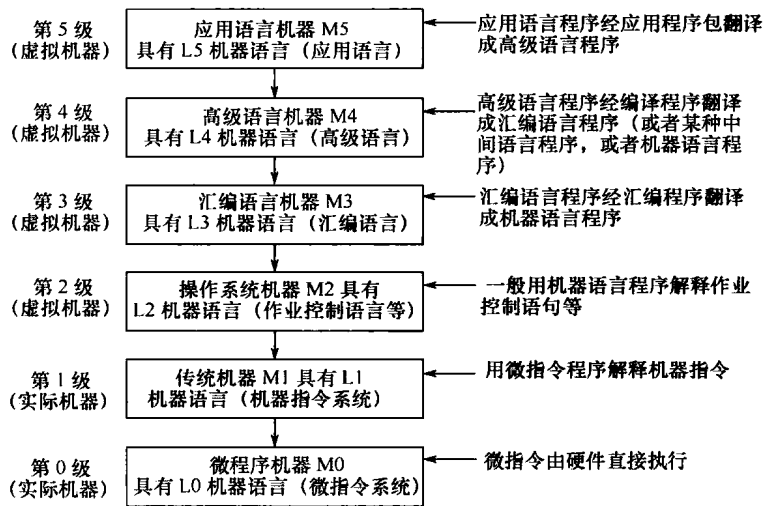


图 1.2 计算机系统的多级层次结构

把计算机系统按功能划分为多个层次结构后,对各机器级的程序员而言,只要熟悉和遵守该级语言的规范准则,所编写的程序就总能在此机器级上运行并得到结果,而不用了解该机器级是如何实现的。各机器级的实现主要靠翻译或解释,或者是这两者的结合。翻译是先用转换程序将高一级机器级上的程序整个地变换成低一级机器级上可运行的等效程序,然后再在低一级机器级上实现的技术。解释则是在低一级机器级上用它的一串语句或指令来仿真高一级机器级上的一条语句或指令的功能,通过对高一级程序中的每条语句或指令逐条解释来实现的技术。

计算机系统采用何种实现方式,要从效率、速度、价格、资源状况、可靠性等多方面因素全盘考虑,对软件、硬件及固件的取舍进行综合平衡。

软件和硬件在逻辑功能上是等效的,同一逻辑功能既可以用软件也可以用硬件或固件实现。从原理上讲,软件实现的功能完全可以用硬件或固件完成;同样,硬件实现的逻辑功能也可以由软件的模拟来完成,只是性能、价格以及实现的难易程度不同而已。例如,在计算机中实现十进制数乘法这一功能,既可以用硬件来实现,也可以用软件来完成。用硬件实现,需要设计十进制数乘法机器指令,其特点是完成这一功能的速度快,但需要更多的器件。而用软件来实现这个功能,则要采用加法、移位等指令通过编程来实现,其特点是实现的速度慢,但不需要增加器件。

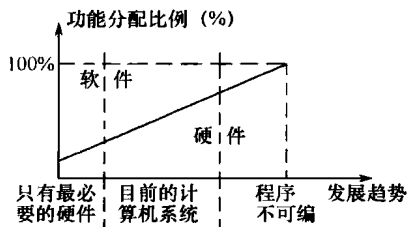


图 1.3 软、硬件功能分配

软、硬件的功能分配比例可以在很宽的范围内变化,这种变化是动态的。如图 1.3 所示,软、硬件功能分配的比例随不同时期以及同一时期的不同机器的变化而变化。由于软、硬件是紧密相关的,软、硬件界面常常是模糊不清的,因此在计算机系统的功能实现上,有时候很难分清哪些功能是由硬件完成的,哪些功能是由软件完成的。在满足应用的前提下,软、硬件功能分配比例的确定主要看能否充分利用硬件、器件技术的现状和进展,使计算机系统达到较高的性能

价格比。对于计算机系统的用户,还要考虑他所直接面对的应用语言所对应的机器级的发展状况。

从目前软硬件技术的发展速度及实现成本上看,随着器件技术的高速发展,特别是半导体集成

技术的高速发展,以前由软件来实现的功能越来越多地由硬件来实现。总的来说,软件硬化是目前计算机系统发展的主要趋势。

1.1.2 按功能划分层次的好处

从概念和功能上把一个复杂的计算机系统看成由多个机器级构成的层次结构,可以有如下的好处。首先,有利于理解软件、硬件和固件在系统中的地位和作用。从系统层次的划分中可以看出,微程序机器级(M0)、传统机器级(M1)、操作系统机器级(M2)不是为应用程序员解题设计的,而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;在这之上的机器级(M3~M5级)则主要是为应用程序设计人员解决各类实际应用问题而设计的。其次,有利于理解各种语言的实质和实现途径。计算机各层次的语言总是通过低一级的语言翻译或解释来实现,这说明相邻机器级之间的语义差别不能太大。再次,有利于推动计算机系统结构的发展。例如,可以重新分配软、硬件的比例,为虚拟机器的各个层次提供更多更好的硬件支持,改变硬件及器件快速发展而软件却日益复杂、开销过大的状况。又如,可以用硬件和固件来实现高级语言和操作系统,从而形成高级语言机器和操作系统机器。再如,用真正的机器来取代各级虚拟机,摆脱各级功能都在同一台实际机器上实现的情况,发展多处理机系统、分布处理系统、计算机网络等系统结构。最后,有利于理解计算机系统结构的定义。把计算机按功能划分成多个不同的层次结构,从各个层次的功能划分和实现去了解计算机系统,有助于更深入地了解系统结构的定义。

1.2 计算机系统结构

“计算机系统结构”这个名词来源于英文 Computer Architecture,也译成“计算机体系结构”。英文 Architecture 这个词原来用于建筑领域,到了 20 世纪 60 年代,这个名词被引入计算机领域。“计算机系统结构”一词已经得到普遍应用,它研究的内容不但涉及计算机硬件,也涉及计算机软件,已经成为一门学科。

1. 计算机系统结构的定义

计算机系统结构这个词是由阿姆达尔(Amdahl)等人在 1964 年介绍 IBM360 时提出的。他们把系统结构定义为由程序设计者所看到的一个计算机系统的属性,即概念性结构和功能特性。这实际上是计算机系统的外特性。按照计算机层次结构,不同程序设计者所看到的计算机具有不同的属性。在计算机技术中,一种本来存在的事物或属性,但从某种角度看却好像不存在,这称为透明性。通常,在一个计算机系统中,低层机器级的概念性结构和功能特性,对高级语言程序员来说是透明的。

例如,使用高级语言的程序员所看到的计算机属性主要是软件子系统和固件子系统的属性,包括程序语言以及操作系统、数据库管理系统、网络软件等用户界面。例如,IBM370 系列机的中央处理器都具有相同的基本指令系统,指令的解释过程在不同档次机器中的处理方式不同,但对程序设计者而言,所看到的数据形式都是相同的 32 位字长。同样,对于不同档次机器,在组成与实现时,数据通路宽度(数据总线)可以分别采用 8 位、16 位、32 位或 64 位,但对于程序员而言,数据总线宽度是透明的,即在设计程序时不需要考虑数据总线宽度的位数。阿姆达尔等人提出的系统结构定义中的程序设计者是指机器语言或编译程序设计者,因此程序设计者所看到的计算机属性,实际上是硬件子系统的概念结构及其功能特性。

2. 计算机系统结构研究的内容

计算机系统结构研究的主要内容是计算机系统中各级之间界面的定义及其上下的功能分配。在这里,我们主要讨论传统机器级的系统结构,即从机器语言程序员的角度所看到的计算机系统结

构,如图 1.4 所示。界面之上的功能由软件实现,界面之下的功能由硬件/固件实现。因此,本书研究软、硬件功能的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、逻辑设计或器件设计等。

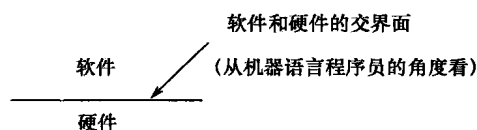


图 1.4 计算机系统软、硬件界面

对于目前的通用型机器,计算机系统结构(传统机器程序员所看到的属性)一般包括以下内容:

- ① 机器内的数据表示,包括硬件能直接识别和处理的数据类型和格式等;
- ② 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;
- ③ 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;
- ④ 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;
- ⑤ 存储系统,包括最小编址单位、编址方式、主存容量和最大可编址空间等;
- ⑥ 中断机构,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;
- ⑦ 机器工作状态的定义及切换,若在管态,则处理机可以使用系统的全部指令和全部资源,而在目态,处理机只能执行用户程序;
- ⑧ 输入/输出机构,包括输入/输出设备的连接方式,处理机/存储器与输入/输出设备间数据传输的方式与格式、传送的数据量以及输入/输出操作的结束与出错标志等;
- ⑨ 信息保护,包括信息保护方式和硬件对信息保护的支持等。

1.3 计算机组成

1. 计算机组成的定义

计算机组成(computer organization,又称计算机设计)是指计算机系统的逻辑实现,包括机器内部的数据流和控制流的组成以及逻辑设计等。它着眼于机器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。

计算机组成的设计是指按所希望达到的性能价格比,最佳、最合理地把各种设备和部件组成计算机,以实现所确定的计算机系统结构。对传统机器程序员来说,计算机组成的设计内容一般是透明的。

2. 计算机组成设计

计算机组成的任务是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,研究各组成部分的内部构造和相互联系,以实现机器指令级的各种功能和特性。这种相互联系包括各功能部件的配置、相互连接和相互作用。各功能部件的性能参数相互匹配是计算机组成合理的重要标志,因而相应地就有许多计算机组织方法。例如,为了使存储器的容量更大、速度更快,人们研究出存储体系。在存储体系中,又包括虚拟存储器、高速缓存、多模块交叉工作、多寄存器组和堆栈等技术。再如,为了使输入/输出设备与处理机间的信息流量达到平衡,人们研究出通道、外围处理机等方式。为了提高处理机速度,人们研究出流水线、多线程、多执行部件等方式。在各功能部件的内部结构研究方面,产生了许多组合逻辑、时序逻辑的高效设计方法和结构。

一般来讲,计算机组成设计要确定以下内容:

- ① 数据通路的宽度,指在数据总线上一次能并行传送的信息位数。

② 专用部件的设置,包括设置哪些专用部件,如乘法专用部件、浮点运算部件、字符处理部件、地址运算部件等,每种专用部件的个数等,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。

③ 各种操作对部件的共享程度,若共享程度太高,则会由于共享部件的分时使用而降低操作的速度;若设置多个同一功能部件以降低共享度,则系统的价格会随之升高。

④ 功能部件的并行度,如功能部件的控制和处理方式采用顺序串行方式,还是采用重叠、流水、分布处理方式。

⑤ 控制机构的组成方式,如控制机构采用硬连线控制还是微程序控制,采用单机处理还是多机处理或功能分布处理。

⑥ 缓冲和排队技术,包括如何在部件间设置多大容量的缓冲器来弥补它们的速度差异,对于等待要求处理的事件如何排队,如随机、先进先出、先进后出、优先级、循环等不同方式。

⑦ 预估、预判技术,如采用何种原则来预测未来的行为,以优化性能和优化处理。

⑧ 可靠性技术,如采用什么样的冗余技术和容错技术来提高可靠性。

1.4 计算机实现

1. 计算机实现的定义

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。计算机实现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。

2. 器件技术研究的内容

器件技术研究的内容包括:处理机、主存等部件的物理结构,器件的集成度和速度,专用器件的设计,信号传输等。例如,集成电路采用 TTL 类型还是 MOS 类型,最小光刻线条的尺寸,连线采用铝连线还是铜连线,集成电路内部的电容、电阻的设计,集成电路上的晶体管采用什么样的材料和工艺,如在不同场合下可以选用硅、硅锗合金、砷化镓等,其带来的物理特性、响应特性、功耗也各有不同。随着时间的推移,芯片的封装技术已经历了好几代的变迁,从 DIP、QFP、PGA、BGA 到 CSP,再到 MCM 等。

计算机所用的基本器件逐步从初期的电子管、晶体管、小规模集成电路(SSI)、中规模集成电路(MSI)、大规模集成电路(LSI)、超大规模集成电路(VLSI),发展到现在的特大规模集成电路(ULSI)和巨大规模集成电路(GSI),其生产工艺从 20 世纪 70 年代的 $10\mu\text{m}$ 发展到现在的 32nm 的水平,只用了不到 40 年的时间,其所能达到的集成度是在单个芯片上集成几十亿个晶体管。例如,在处理器方面,Intel 公司的第一个 CPU 的集成度为 2000 多个晶体管;到了 2006 年,Intel 公司推出的 Conroe Core 2 Duo(双核)使用 45nm 制作工艺,在单个芯片上集成了 2.91 亿个晶体管;2007 年,AMD 公司推出的 K10 Barcelona Opteron(四核)使用 65nm 制作工艺,在单个芯片上集成了 4.63 亿个晶体管;2010 年,Intel 推出了制作工艺为 32nm 的 Westmere-EP(六核),单个芯片上集成了多达 11.7 亿个晶体管。

计算机所用的基本器件不仅由分立器件发展到集成电路,而且其速度也有显著提高。器件的发展是推动系统结构和组成前进的关键因素和主要动力。随着生产工艺以及材料工艺的发展,单个器件的尺寸可以越来越小,单个晶体管的尺寸已经从微米级缩小到纳米级,响应速度可以达到 100GHz 的量级,例如,IBM 使用硅锗材料制造的晶体管,速度达到 350GHz,为 CPU 的主频达到 10GHz 的水平提供了技术保障。器件技术的快速发展,使单个器件或集成电路的性能越来越高,而

价格则越来越低,同时,也为各种新的系统结构和组成实现提供了硬件支持,并使计算机的应用得到快速普及。

器件技术的发展还促进了算法、语言和软件的发展。随着 CPU 性价比的迅速改善,加速了大规模高性能并行处理机和通信网络的发展。由数百甚至上万个微处理器组成的 MPP 系统促使人们不断研究新的并行算法、并行语言及相关应用软件和操作系统,以使系统的规模和处理速度能随结点处理器数的增加而显著提高。在 2010 年 11 月 14 日全球超级计算机 500 强排行榜上,经过技术升级的中国“天河一号”超级计算机超过美国首次成为全球运行速度最快的超级计算机。全新升级后的“天河一号”基于 NUDT YH Cluster 集群,硬件上配备了 Intel Xeon X5670 2.93GHz 六核心处理器(制作工艺为 32nm 的 Westmere-EP)、我国自主研发的飞腾 FT-1000 八核心处理器、NVIDIA Tesla M2050 高性能计算卡、224TB 内存、专有互连架构、Linux 操作系统,总计 186368 个核心,持续性能为 2.566PFLOPS(每秒千万亿次浮点运算),峰值性能达 4.701PFLOPS。

3. 系统结构、组成和实现三者的关系与相互影响

计算机系统结构、计算机组成和计算机实现是三个互不相同的概念,但它们是互相联系和相互影响的。计算机系统结构是计算机系统的软、硬件分界面,是机器语言程序设计者或者编译程序生成系统为使其设计或生成的程序能在机器上运行所看到的计算机属性;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。

具有相同系统结构的计算机可以采用不同的组成。例如,具有相同指令系统的计算机,指令的取出、译码、取操作数、运算、存结果既可以采用顺序方式进行解释,也可以采用流水方式让它们在时间上重叠进行来提高速度。又如,乘法指令可以利用专用乘法器来实现,也可以通过加法器重复相加、移位来实现,这主要取决于对速度的要求、乘法指令出现的频度和所采用的乘法运算方法。显然,前一种方法可以有效地提高乘法运算速度,而后一种方法则可以降低系统的价格。

一种系统结构可以有多种计算机组成,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,在 CPU 的选择上,同一类型的 CPU 一般都有多个不同的工作频率,从速度及价格等角度考虑,可以选择工作频率较高的 CPU 来实现较高的运算速度,也可以选择工作频率较低的 CPU 来保证系统的价格相对较低。又如,在主存器件的选择上,可以选择 TTL 型的器件,也可以采用 MOS 型器件;既可以采用单片 VLSI 集成电路,也可以采用多片 LSI 或 MSI 集成电路组成;既可以选择响应速度较快的芯片,也可以选择响应速度较慢的芯片。实际上,这也是在速度、价格等因素之间进行选择。换句话说,采用什么样的实现技术主要考虑所要达到的性能价格比及器件技术的现状。

反过来,计算机实现是计算机系统结构和计算机组成的基础,其技术的发展对计算机系统结构有着很大的影响,而计算机组成也会影响计算机系统结构。例如,器件的发展使系统结构由大型机下移到小型机及微机中的速度加快,早期用于大型机的各种数据表示、指令系统、操作系统很快被应用到小型机及微机中。例如,在目前的 PC 机中使用的 CPU,都普遍采用了在早期大型机中才使用的超标量技术,并引入了 VLIW 技术,有些机器还使用了超流水线技术。

系统结构的设计必须结合应用考虑,为软件和算法的实现提供更多、更好的支持。同时,还要涉及可能采用和准备采用的组成技术。即计算机系统结构的设计应考虑减少对各种组成及实现技术的使用限制,在一种系统结构中,应允许有多种不同的组成和实现技术,既能方便地在低档机器上用简单、低成本的组成实现,也能在高档机器上以较高的成本、复杂的组成实现。例如,在 IBM370 系列机中,由低到高有不同档次的机器,它们的中央处理器都具有相同的基本指令系统,只是指令的分析、执行方式不同,可以选择在低档机器上用顺序方式处理,在高档机器上用并行方式处理。又如,在数据通路宽度的组成和实现上,不同档次机器可以分别采用 8 位、16 位、32 位和 64 位。IBM370 系列机采用通道方式进行输入/输出,可以选择在低档机器中采用的结合型通道和在高档机器中采用的独立型通道。

在不同时期,系统结构、组成和实现所包含的内容会有所不同,在某些计算机系统中作为系统结构的内容,在另一些计算机系统中可能是组成和实现的内容。随着各种新技术的出现和发展,特别是器件技术的发展,各种功能都集成到一个芯片上,使系统结构、组成和实现融合于一体,系统结构、组成和实现三者之间的界限变得越来越模糊。

1.5 计算机性能

计算机系统的性能指标主要取决于计算机的本质属性,即时空属性。时间是衡量计算机性能的标准,对于同样的工作量,花费的时间越少,速度就越快。衡量机器性能的唯一固定而且可靠的标准就是真正执行程序的时间。

1.5.1 性能因子 CPI

CPU 的工作是建立在一个固定的时钟节拍上的,这个节拍有一定的时间长度 T_c ,我们称之为处理器时钟周期,简称时钟周期(clock cycles)。它是计算机内部操作的基本时间单位,即计算机内部每一项基本功能操作在一个或多个时钟周期的时间内完成。时钟周期的单位通常使用 ns(纳秒),它的倒数 f_c 则是时钟的频率,即 CPU 的主频,通常使用 MHz 或 GHz 作为单位。

一个程序在 CPU 上运行所需的时间 T_{CPU} 可以用下述公式表示

$$T_{CPU} = I_N \times CPI \times T_c \quad (1.1)$$

在式(1.1)中, I_N 表示要执行程序中的指令总数, CPI 表示执行每条指令所需的平均时钟周期数(Clock cycles Per Instruction, CPI)。由此公式可见,程序运行的时间取决于三个特征:时钟周期(或速率),每条指令所需的时钟周期数,以及程序中总的指令数。由式(1.1)可得到 CPI 的表达式

$$CPI = \frac{\text{执行整个程序所需处理器时钟周期数}}{\text{程序中指令总数}} \quad (1.2)$$

在程序执行过程中,要用到不同类型的指令。令 I_i 表示第 i 类指令在程序中执行的次数, CPI_i 表示执行一条第 i 类指令所需的时钟周期数, n 为程序中所有的指令种类数,则式(1.2)可以改写为

$$CPI = \frac{\sum_{i=1}^n CPI_i \times I_i}{I_N} \quad (1.3)$$

因为 I_N 是一个常数,所以式(1.3)可以改写为

$$CPI = \sum_{i=1}^n (CPI_i \times I_i / I_N) \quad (1.4)$$

其中, I_i / I_N 表示第 i 类指令在程序中所占的比例,又称指令混合比。式(1.4)说明,平均 CPI(或称有效 CPI)等于每类指令的 CPI 和该类指令在整个程序中出现的百分比的乘积之和。在这里,可以用 CPU 时间来评价 CPU 性能,即

$$\text{CPU 性能} = (I_N \times CPI \times T_c)^{-1} = \frac{f_c}{I_N \times CPI} \quad (1.5)$$

由式(1.5)可以看出,要改善 CPU 性能,可以通过改变其中某一项或几项因素来实现。事实上,这三个因素是相互关联的。时钟频率 f_c 由硬件技术和组成技术决定, CPI 与组成和指令集有关,而指令数 I_N 则与指令集和编译技术有关。

1.5.2 计算机性能常用指标 MIPS 和 MFLOPS

1. MIPS(Million Instructions Per Second,百万条指令每秒)

MIPS 是一个用来描述计算机性能的尺度。对于一个给定的程序, MIPS 定义为