

● 李 勇 刘恩林 编著



计算机体系结构

● 国防科技大学出版社

● 李 勇 刘恩林 编著

计算机体系结构

● JISUANJI TIXIJIEGOU

国防科技大学出版社

内 容 简 介

本书讲述计算机体系结构的基本概念、基本原理、基本结构和基本分析方法。对近年来出现的一些较成熟的新概念和新技术也加以阐述和介绍。每章内容都安排1~2个典型的计算机体系结构,以进行深入的介绍和分析,如IBM370、VAX-11/780、银河巨型机(YH-1)、CDC CYBER-205、BSP、C_m*等,使内容更加丰富、具体。

本书内容是从事计算机系统设计和软件设计所必须具备的专业知识。内容充实、取材新颖、层次结构清楚、语言简练,便于读者自学,也便于教师因材施教。它适于作计算机专业本科生和有关专业研究生的教材,也可作为计算机科技工作者的参考书。

计 算 机 体 系 结 构

李 勇 编著
刘恩林

国防科技大学出版社 出版发行

湖南省新华书店经销
国防科技大学印刷厂印装

开本: 787×1092 1/16 印张: 22¹²/₁₆ 字数: 518千
1988年8月第1版 1988年8月第1次印刷 印数: 1—20 000

ISBN 7-81024-037-4

平装

TP·9 定价: 4.15元

前 言

本书是作为计算机专业《计算机体系结构》教材，经过两期教学试用，又经过认真地修改和充实后定稿出版的。

本书内容是根据统编教材的教学大纲和计算机发展需要而确定的。作者力求深入浅出地阐述计算机体系结构的基本概念、基本原理、基本结构和基本分析方法，并介绍了近年来出现的一些较成熟的新概念和新技术。为充分反映现代计算机发展水平，在深入讨论单机体系结构的基础上，加强了并行性概念和并行结构的介绍和性能分析。

该书吸收了作者多年从事计算机体系结构的教学经验和科研实践成果，力图使内容丰富、具体，既便于教员因材施教，也便于读者自学。每章内容都安排1~2个典型实例加以介绍和分析。一些有代表性的计算机，如IBM370、VAX—11/780、银河巨型机（YH—1）、CDC CYBER-205、BSP、C_m等，对其体系结构都进行了较深入地介绍和分析。每章后面都附有一定数量的习题。

全书共包括八章。它们是：计算机体系结构的基本概念、计算机体系结构设计技术、存贮与I/O子系统、标量处理机、向量处理机、并行处理机、多处理机、计算机体系结构的新发展等。由于《容错与诊断》、《纠错码》、《计算机系统性能评价技术》单独设课。因此本书不包括这些内容。在编写和内容衔接上，作者力求语言简炼、层次结构清楚。

本书由李勇同志负责编写第2、3章，并改写了第8章；刘恩林同志负责编写第1、4、5、6、7、8章，并改写第2、3章。全书由刘恩林整理、李勇审定。

本书出版得到了国防科大出版社的大力支持，也得到许多兄弟院校的支持和鼓励，在书稿的抄写和绘图工作中，又得到了我校84级许多同学的支持和帮助，在此一并致谢。特别要感谢本书的编辑宋焕章同志，他为本书的出版作了大量的工作。

由于作者学识有限，谬误之处在所难免，敬请读者批评指正。

编著者

1988年2月，长沙

目 录

1. 计算机体系结构的基本概念

1.1 现代计算机系统的多级层次结构	(1)
1.2 计算机的体系结构、组成和实现	(4)
1.2.1 计算机体系结构	(4)
1.2.2 计算机组成	(6)
1.2.3 计算机实现	(7)
1.2.4 体系结构、组成和实现三者的关系	(7)
1.3 计算机体系结构和并行性概念的发展	(11)
1.3.1 Von Neumann 型计算机体系结构及其发展	(12)
1.3.2 非 Von Neumann 型计算机	(17)
1.3.3 体系结构中并行性概念的发展	(19)
1.3.4 频带平衡的概念	(27)
1.4 计算机系统的分类	(28)
1.4.1 弗林分类法	(29)
1.4.2 冯氏分类法	(31)
习 题	(32)

2. 计算机体系结构设计技术

2.1 引 言	(35)
2.2 体系结构的设计方法	(37)
2.2.1 计算机系统的设计方法	(37)
2.2.2 分级设计	(41)
2.2.3 试探性设计	(42)
2.3 体系结构的描述	(42)
2.3.1 PMS表示法	(43)
2.3.2 结构表示法ASN	(43)
2.4 主要软、硬件界面的确定	(50)
2.4.1 数据表示	(50)
2.4.2 寻址技术	(54)
2.4.3 指令系统	(60)
2.4.4 中断系统	(81)
2.5 性能评价技术	(87)
2.5.1 概 述	(87)
2.5.2 确定性模型	(88)

2.5.3 排队模型	(90)
------------	------

2.5.4 模拟模型	(93)
------------	------

习 题	(95)
-----	------

3. 存储器 and I/O 子系统

3.1 存储器的层次结构	(99)
3.1.1 单级存储器的基本矛盾和解决办法	(99)
3.1.2 存储器的层次结构和性能参数	(101)
3.2 虚拟存储器	(104)
3.2.1 虚拟存储器概念	(104)
3.2.2 程序的局部性与定位	(105)
3.2.3 页式虚拟存储器系统	(107)
3.2.4 段式虚拟存储器系统	(121)
3.2.5 段页式虚拟存储器系统	(123)
3.3 高速缓冲存储器(Cache)	(125)
3.3.1 Cache的结构原理与特点	(125)
3.3.2 Cache的地址映射与变换	(126)
3.3.3 替换算法的硬件实现	(133)
3.3.4 Cache存储器的某些性能分析	(135)
3.4 Cache—主存—辅存三级层次	(139)
3.5 I/O 子系统	(140)
3.5.1 概 述	(140)
3.5.2 I/O 处理机(通道)方式	(141)
3.5.3 外围处理机方式	(145)
习 题	(148)

4. 标量处理机

4.1 一次重迭	(153)
4.1.1 基本原理和结构	(153)
4.1.2 一次重迭方式设计中的若干问题	(156)
4.2 先行控制	(161)
4.2.1 基本原理和结构	(161)
4.2.2 先行控制方式设计中的若干问题	(164)
4.3 多操作部件处理机	(169)
4.3.1 基本结构	(170)

I

4.3.2 多操作部件处理机设计中的若干问题.....	(173)	6.5.1 BSP的设计思想和特性.....	(272)
4.4 流水线处理机.....	(178)	6.5.2 BSP的结构.....	(274)
4.4.1 基本概念.....	(178)	6.5.3 BSP的向量指令及其实现.....	(277)
4.4.2 流水线结构.....	(188)	习题.....	(280)
4.4.3 流水线处理机设计中的若干问题.....	(194)		
习题.....	(205)		
5. 向量处理机		7. 多处理机	
5.1 向量处理方式.....	(208)	7.1 多处理机结构与特点.....	(283)
5.1.1 水平(横向)处理方式.....	(208)	7.1.1 多处理机结构.....	(283)
5.1.2 垂直(纵向)处理方式.....	(209)	7.1.2 多处理机系统的特点.....	(286)
5.1.3 分组(纵横)处理方式.....	(210)	7.2 多处理机系统的硬件组成.....	(287)
5.2 银河YH-1向量处理机.....	(211)	7.2.1 互连网络.....	(287)
5.2.1 系统组成.....	(211)	7.2.2 处理机特性.....	(296)
5.2.2 中央处理机结构.....	(213)	7.2.3 存储器组成.....	(297)
5.2.3 YH-1机的向量处理.....	(220)	7.3 多处理机系统的软件组成.....	(302)
5.2.4 YH-1系统特点.....	(225)	7.3.1 程序中并行性的发掘.....	(302)
5.3 CDC CYBER-205向量处理机.....	(228)	7.3.2 多处理机的操作系统.....	(312)
5.3.1 中央处理机结构.....	(229)	7.4 多处理机结构举例.....	(316)
5.3.2 CYBER-205的向量处理.....	(232)	7.4.1 C_m^* 结构.....	(316)
5.4 向量处理机的性能分析.....	(235)	7.4.2 C_{mmp} 结构.....	(319)
5.4.1 向量处理机与标量流水处理机的性能比较.....	(236)	习题.....	(321)
5.4.2 不同类型向量处理机的性能分析.....	(237)		
习题.....	(242)	8. 计算机体系结构新发展	
6. 并行处理机		8.1 引言.....	(322)
6.1 并行处理机的结构与性能分析.....	(244)	8.2 缩减指令系统计算机.....	(323)
6.1.1 并行处理机的典型结构.....	(244)	8.2.1 CISC与RISC.....	(323)
6.1.2 并行处理机的性能分析.....	(245)	8.2.2 RISC体系结构特点.....	(325)
6.1.3 并行处理机特点.....	(247)	8.2.3 RISC编译程序特点.....	(328)
6.2 并行处理机的算法举例.....	(248)	8.3 数据流计算机.....	(331)
6.2.1 阵列处理机简介.....	(248)	8.3.1 控制流计算与数据流计算.....	(331)
6.2.2 算法举例.....	(249)	8.3.2 数据流程图.....	(335)
6.3 SIMD计算机的互连网络.....	(254)	8.3.3 数据结构.....	(338)
6.3.1 引言.....	(254)	8.3.4 数据流语言.....	(340)
6.3.2 基本的互连函数.....	(255)	8.3.5 静态数据流计算机体系结构.....	(341)
6.3.3 单级互连网络及其特性.....	(259)	8.3.6 动态数据流计算机体系结构.....	(344)
6.3.4 多级互连网络及其实现.....	(262)	8.4 归约机.....	(347)
6.4 ICL的分布式阵列处理机DAP.....	(267)	8.4.1 引言.....	(347)
6.4.1 引言.....	(267)	8.4.2 函数式语言.....	(347)
6.4.2 DAP的结构特点.....	(269)	8.4.3 归约机的特点与表达式归约.....	(348)
6.4.3 处理方式和指令系统.....	(271)	8.4.4 图归约机.....	(350)
6.5 Burroughs科学处理机BSP.....	(272)	8.4.5 串归约机.....	(353)
		8.5 展望.....	(355)
		习题.....	(357)
		参考文献	

1 计算机体系结构的基本概念

自从1946年第一台电子数字计算机问世以来,计算机的发展先后经历了四代,目前正向第五代过渡。人们通常习惯以电子管(V)、晶体管(T)、中小规模集成电路(MSI、SSI)、大规模集成电路(LSI)、超大规模集成电路(VLSI)等器件的变革作为计算机换代的标志。当然,这种器件的更新,其速度、功能、可靠性的不断提高和价格的不断降低,确实在不断地推动着计算机的发展,是计算机性能得以提高的物质基础。但是,如果把某一时期器件性能的提高(如用器件级延迟时间表示)和同一时期的计算机系统性能的提高(如用平均指令时间表示)作一番比较,不难发现:计算机系统性能提高的指数要比器件级性能提高的指数大得多。P·H Enslow曾统计了1965~1975这十年间,器件级延迟时间和计算机系统平均指令时间随时间变化的情况,结果表明:这十年间器件性能提高了近十倍,然而计算机系统的性能提高了近百倍。这种情况在计算机近年来的发展中变得更加明显。这说明计算机系统性能提高除了器件这个最根本的因素外,还有其它多种因素。其中计算机体系结构的改进是提高计算机系统性能的重要因素。

那么,什么是计算机体系结构(Architecture)?计算机系统的层次结构又指什么?体系结构、计算机组成和实现三者之间有什么关系?近四十年来,计算机体系结构和并行性概念有哪些发展?怎样从体系结构角度对计算机系统进行分类?本章就是要解决上述问题,以使读者在学习后续各章之前,对计算机体系结构有一个基本的认识。

1.1 现代计算机系统的多级层次结构

现代计算机系统是由软件和硬/固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不同的观点和方法。C.G Bell等人列举了计算机系统的七种观点。本节从使用语言的角度,把计算机系统按功能划分成多级层次结构。

随着计算机系统的发展,已设计出一系列语言。从面向机器的语言(如指令系统、汇编语言),到各种面向过程的高级程序设计语言(如FORTRAN、PASCAL),到各种面向问题的语言或者叫应用语言(如面向人工智能的语言PROLOG)。计算机语言就是这样由低级向高级发展,每一种语言都比它的前一种语言功能更强,更便于人们应用,但又都以它的前一种语言为基础。

计算机语言可分成一系列的层次或级(Level),最低层语言的功能最简单,最高层语言的功能最强。对于用某一层语言编写程序的程序员来说,他一般是不管其程序在机

器中是如何执行的，只要程序正确，他终能得到预期的结果。这样，对这层语言的程序员来说，他似乎有了一种新的机器，这层语言就是这种机器的机器语言，该机器能执行用该层语言写的全部程序。因此计算机系统就可以按语言的功能划分成多级层次结构。每一层以一种不同的语言为特征。尽管在各层内部还可能细分成子层，又可能构成局部的层次结构，但只要它们是用同一种语言描述的，它们就不构成单独的层和级。这样，可以把现代计算机系统画成如图 1.1 所示的层次结构。在图中我们称由软件实现的机器为虚拟机器 (Virtual Machine)，以区别于由硬件或固件实现的实际机器。

第 1 级是微程序机器级。这级的机器语言是微指令系统，程序员用微指令编写的微程序一般是直接由硬件解释实现的。但近年来又有人提出毫微程序设计的方法，这时每一条微指令由一段毫微程序解释实现，如图中第 0 级所示。由于毫微程序设计的机器还未得到广泛应用，故在图中用虚线画出，但由此可以说明多层结构还可以从微程序机器级向下延伸。

第 2 级是传统机器级。这级的机器语言是该机的指令系统，程序员用这级指令系统编写的程序是由微程序进行解释的，这个解释程序运行在第 1 级上。由微程序解释指令系统又称作仿真 (Emulation)。实际上，在第 1 级可以有一个或数个能够在它上面运行的解释程序，每一个解释程序都定义了一种指令系统。因此，可以通过仿真在一台机器上实现多种指令系统。

必须指出：在一些大型计算机系统中 (如 CRAY-1) 并没有微程序机器级，在这些计算机系统中是用硬件直接实现传统机器的指令系统，而不必由任何解释程序进行干预。

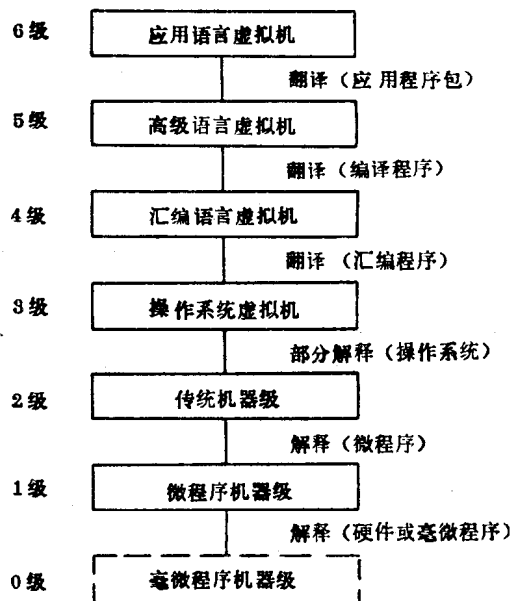


图 1.1 现代计算机系统的多级层次结构

第 3 级是操作系统虚拟机。在按功能把计算机系统划分成多级层次结构时，把操作

系统置于哪一层是有争论的。多数人认为：从操作系统的基本功能来看，一方面它要直接管理传统机器中的软硬件资源，另一方面它又是传统机器的引伸。它提供了传统机器所没有的、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，存贮体系和多道程序以及多重处理所用的某些操作等等。因此，把操作系统机器级置于传统机器级之上和汇编语言机器级之下是比较合理的。

A. S. Tanenbaum 根据这一级所用语言的特点，把该级又称为混合级。这是因为这级语言中的多数指令就是第 2 级传统机器的指令，如算术运算，逻辑运算和移位等类型指令，这些指令称之为“常规的”第 3 级指令。此外，在这一级中还提供了传统机器所没有的另一部分指令，用以支持汇编语言和高级语言的使用和实现，如文件管理中的打开文件、读/写文件、关闭文件等指令，使在第 3 级上工作的程序员在进行 I/O 操作时，不必再使用第 2 级上那些繁琐的 I/O 指令，这些指令称之为 OSML (Operating System Machine Level) 指令。

程序员用这一级语言所编写的程序，一部分是由第 2 级进行解释的。运行在第 2 级上的解释程序，习惯上就叫作操作系统。而那些与第 2 级指令相同的第 3 级指令是直接由微程序实现的。换句话说，OSML 指令是由操作系统进行解释的，而那些常规的第 3 级指令是由微程序进行解释的。

由于 OSML 指令是由软件解释实现的，因此这一级称为操作系统虚拟机。

第 4 级是汇编语言虚拟机。这级的机器语言是汇编语言，用汇编语言编写的程序，首先翻译成第 3 级或第 2 级语言，然后再由相应的机器进行解释。完成翻译的程序就叫做汇编程序。

第 4 级上出现了一个重要变化。通常的第 1、2 和 3 级是用解释 (Interpretation) 方法实现的，而第 4 级或更高级则经常是用翻译 (Translation) 方法实现。

翻译和解释是语言实现的两种基本技术。在这两种技术中，虽然最终都是以执行一串 N 级指令来实现 N+1 级指令，但二者仍存在着差别：翻译技术是先把 N+1 级程序全部变换成 N 级程序后，再去执行新产生的 N 级程序，在执行过程中 N+1 级程序不再被访问。而解释技术是每当一条 N+1 级指令被译码后，就直接去执行一串等效的 N 级指令，然后再去取下一条 N+1 级的指令，依此重复进行。在这个过程中不产生翻译出来的程序，因此解释过程是边变换边执行的过程。在实现新的虚拟机器时，这两种技术都被广泛使用。一般来说，解释比翻译花费的时间较多，但存贮空间占用较少。

第 5 级是高级语言虚拟机。这级的机器语言就是各种高级语言，目前高级语言已达数百种。用这些语言所编写的程序一般是由称为编译程序的翻译程序翻译到第 4 级或第 3 级上，个别的高级语言也用解释的方法实现。

第 6 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，因此这一级语言就是各种面向问题的应用语言。可以设计专门用于人工智能、教育、行政管理、计算机设计等方面的虚拟机，这些虚拟机是当代计算机应用领域的重要研究课题。应用语言编写的程序一般是由应用程序包翻译到第 5 级上。

要注意，各层机器的实现和各层机器上的题目得到解答不是一回事。后者要从你用

的那级开始逐级变换，直至到达最低一级，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一级语言的程序，或是由低一级的程序所解释。

把复杂的计算机系统按功能划分成多级层次结构有明显的优点。首先，它有利于我们正确地理解计算机系统的工作，明确软件、硬件和固件在计算机系统中的地位和作用。事实上最低的三级并不是为一般程序人员求解问题而设计的，而主要是为了运行支持更高层次所必须的解释程序和翻译程序而设计的。这些解释程序和翻译程序是由系统程序设计人员所编写的，他们专门设计和实现新的虚拟机。而高三级是为应用程序设计人员求解问题用的。按此分层观点，对于理解像分布式系统、计算机网络和多机系统等更复杂的计算机系统的工作也是有好处的。

其次，这种多级层次结构对于理解各种语言的实质及其实现也是有好处的。计算机语言的发展是愈来愈便于人们的应用，因而更接近于人的自然语言，但它总是由低一级的语言翻译或解释实现的。这就告诉我们，为了便于翻译和解释，相邻级的语言的语义结构差别不能过大。

再次，这种多级层次结构对于探索虚拟机新的实现方法、设计新的计算机系统也是有好处的。在设计新的计算机系统时，人们一方面努力提高传统机器级的语义以缩小这种语义差距；另一方面，让某些级的程序由更低级进行解释，或者直接翻译成更低级的程序。例如：由微程序或硬件直接实现高级语言的所谓高级语言计算机，由固件直接实现操作系统的所谓操作系统计算机就是这方面的代表。当然，采用何种实现方式，要从整个计算机系统的效率、速度、造价及资源使用状况等方面综合考虑，以便提供更高的性能价格比。

最后，这种多级层次结构有助于我们理解计算机体系结构的定义，从而更合理地进行计算机系统设计。

1.2 计算机的体系结构、组成和实现

计算机体系结构(Computer Architecture)这个词目前已被广泛使用。Architecture本来用在建筑方面，译为“建筑学、建筑术、建筑样式、构造、结构”等。这个词被引入计算机领域后，最初的译法也各有不同，以后趋向译为“体系结构”，但关于它的定义仍未统一。本节在G.M Amdahl提出的体系结构定义的基础上，分析体系结构、组成和实现三者的关系。

1.2.1 计算机体系结构

1964年，Amdahl在介绍IBM360系统时提出：计算机体系结构是程序员所看到的计算机的属性，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所具有的功能特性。显然，不同的计算机系统，如图1.2所示的NOVA机和PDP—11机，DJS—1000和DJS—2000机，从传统机器级程序员或汇编语言程序员看，是具有不同的属性的。但是，从高

级语言（假如它们都配置某种高级语言如FORTRAN）程序员看，它们就几乎没有什么差别，具有相同的属性。或者说，这些传统机器级所存在的差别是高级语言程序员所“看不见”的，是不需要他们知道的。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象不存在的概念称为透明性（Transparency）。通常，在一个计算机系统中，低层机器的属性往往对高层机器的程序员是透明的，如传统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。由此看出，在层次结构的各个级上都有它的体系结构。Amdahl提出的体系结构是指传统机器级的体系结构，即一般所说的机器语言程序员所看到的传统机器级所具有的属性。

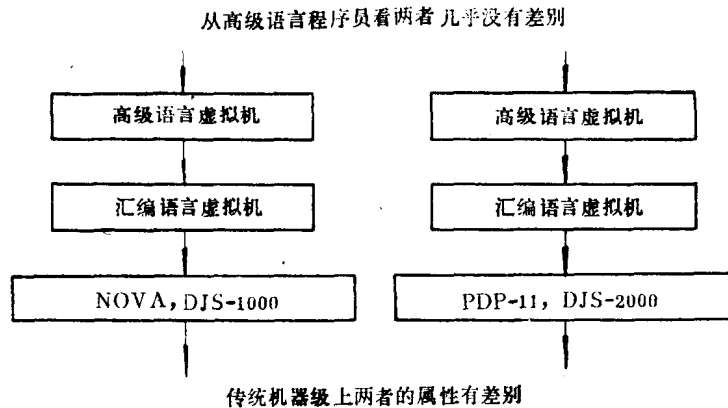


图 1.2 透明性概念的理解

这些属性是机器语言程序设计者（或者编译程序生成系统）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，这些属性包含其概念性结构和功能特性两个方面。显然，这些属性不只是与处理机有关，而应是工作于传统机器上的程序员所看到的机器的所有部分，包括处理机、存贮系统、I/O 联结方式和中断机构等。但是，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。

目前，对于通用寄存器型机器，这些属性主要是指：

- 数据表示（硬件能直接辨认和处理哪些数据类型）；
- 寻址规则（包括最小寻址单元、寻址方式和寻址方式的表示）；
- 寄存器定义（包括操作数寄存器、变址寄存器和控制寄存器等定义、数量和使用方式）；
- 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）；
- 中断机构（中断的类型和中断响应硬件的功能等）；
- 机器工作状态（如管态和目态等）的定义和切换；
- 存贮系统（主存容量、程序员可用的最大存贮容量等）；
- 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联结方式、处理机/存贮器与 I/O 设备间数据传送的方式和格式、传送的数据量、以及 I/O 操作的结束与出错指示等），等等。

由此看出，这些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出在传统机器上正确运行的程序。因此，计算机体系结构的实质是计算机系统中软硬件界面的确定。如图 1.3 所示，其界面之上的功能是所有软件的功能，界面之下的功能是所有硬件和固件的功能。研究计算机体系结构就是研究计算机系统中软、硬件功能的分配。

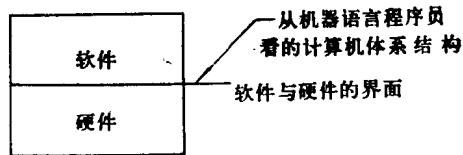


图 1.3 计算机体系结构的实质

对汇编语言程序员来说，这些属性的大部分也是应该看到的，但对高级语言程序员来说却是透明的。

1.2.2 计算机组成

计算机组成 (Computer Organization) 有时也称计算机设计 (Computer Design)。它要确定实现体系结构特性所需的操作部件的性质。在 CPU 部件中就是要确定各事件的排序方式和控制机构，能同时处理的数据量，以及 CPU 与系统的其它部件之间的功能划分。

我们可以从计算机体系结构和计算机组成的对比来认识它。例如，指令系统的确定属计算机体系结构，而指令的实现，如取指令、取操作数、运算、存结果等事件的操作及其排序则属于计算机组成。有的机器这四个操作是按顺序方式执行的，而有的机器为了提高速度则按重迭或流水方式执行。又如，决定是否有乘法指令属计算机体系结构，但乘法指令是由专用乘法器实现，还是经加法器用重复相加和移位操作实现，则属计算机组成。这主要取决于计算机所要达到的速度，以及在程序中乘法指令出现的频度和所采用的乘法运算方法。再如，对主存系统，主存的容量和寻址方式的确定属计算机体系结构，而为要达到所定的性能价格比，主存的速度应多高，在逻辑结构上需要采用什么措施（如多体交叉存贮器），则属计算机组成，等等。

由此可见，所谓计算机组成是指计算机体系结构的逻辑实现，包括机器内部的数据流和控制流的组成以及逻辑设计等。计算机组成的设计是按所希望达到的性能价格比，如何最佳、最合理地把各种设备和部件组成计算机，以实现所定的计算机体系结构。

一般计算机组成设计要确定的内容有：

- 数据通路宽度（指处理机中信息能并行传送的位数）；
- 各种操作对功能部件的共享程度（如果使不同功能的操作共享一个部件，即使这些功能在逻辑上互不相关，也只能顺序地执行。为了提高速度，则可重复设置多个这样的部件，使这些功能并行执行，但将带来价格的提高）；
- 专用功能部件的设置（设置具有何种功能的专用部件，如乘（除）法专用部件、浮点运算部件、地址运算部件等，每种专用功能部件设置数量的多少，都取决于所需达到的机器速度，专用部件的使用频度和所需价格等）；

- 功能部件的并行性（指功能部件的控制与处理方式，是否采用重迭或流水技术来提高部件的并行性）；
- 缓冲和排队技术（缓冲技术要确定如何设置缓冲器，以弥补不同部件之间速度的差距，缓冲器容量的确定等；排队技术要确定对等待处理的事件如何安排其处理次序，如先进先出，后进先出、优先数法等）；
- 预测技术（根据系统在过去近期的行为猜测系统在未来近期要干什么，根据这种猜测可优化数据流）；
- 可靠性技术（采用何种硬件冗余技术提高计算机的可靠性）；
- 控制机构的组成（指事件或操作的排序机构是采用硬联技术，还是采用微程序设计技术），等等。

由上述可见，计算机组成设计的内容，对传统机器程序员来讲一般是透明的。

1.2.3 计算机实现

计算机实现 (Computer Implementation) 指的是计算机组成的物理实现。它包括的主要内容有处理机、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；此外还包括电源、冷却、微组装技术和整机装配技术等。

我们还是用前述计算机组成中的例子来说明计算机的实现。例如，乘法指令的选定属计算机体系结构；但乘法指令是用专用乘法器实现，还是经加法器用重复的相加和移位操作实现，则属计算机组成；而乘法器或加法器的物理实现，如器件的选定，电路设计和微组装技术等则属计算机实现。又如对主存系统，主存的容量和寻址方式的确定属计算机体系结构；主存的速度应多高，在逻辑结构上采用什么措施属计算机组成；而主存的物理实现，如存贮器采用什么样的器件、存贮器的逻辑电路设计和微组装技术则属计算机实现。

计算机实现涉及的内容本书不再论述。需要强调的是：计算机实现是计算机体系结构和组成的基础。先进的计算机实现技术，尤其是器件技术的发展对计算机体系结构和组成产生了重大的影响，它一直是推动计算机体系结构和组成发展的最活跃的因素。

1.2.4 体系结构、组成和实现三者的关系

体系结构、组成和实现是三个不同的概念。体系结构是计算机系统的软、硬件的界面，计算机组成是计算机体系结构的逻辑实现，而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，但又有紧密的关系。

我们以系列机 (Family Machine) 为例说明这三者之间的关系。计算机体系结构这个名词，是Amdahl等人在1964年研制IBM360系列机时明确提出的。系列机的出现被认为是计算机发展史上一个重要里程碑。直到现在，各计算机厂家仍按系列机的思想发展自己的计算机产业。如IBM公司在360系列机后，又相继推出IBM370、303X、4300、308X、309X等系列；DEC公司在PDP-11系列后，又推出VAX-11系列；CDC公司的6600、7600、CYBER等系列；CRAY公司的CRAY巨型机系列等等。

所谓系列机就是指在一个厂家内生产的具有相同的体系结构，但具有不同组成和实现的一系列不同型号的机器。如IBM370系列机有370/115、125、135、145、158、168

等一系列从低速到高速的各种型号。它们各有不同的性能和价格，采用不同的组成和实现技术，但从程序设计者所看到的机器属性却是相同的。如它们都具有图 1.4 所示的中央处理机——通道——设备控制器——设备四级连接的概念性结构。即它们在概念上都是采用通道方式进行输入输出，但在计算机组成上，低档机器可以把通道合并到中央处理机，用一套硬件设备分时执行中央处理机和通道的功能（结合型通道），而在高档机上则有独立的通道硬件（独立型通道）。

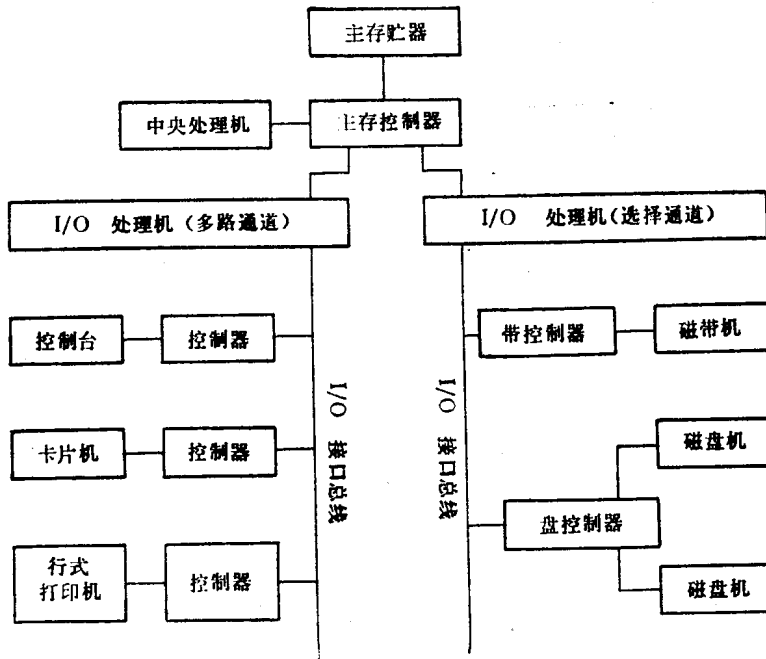
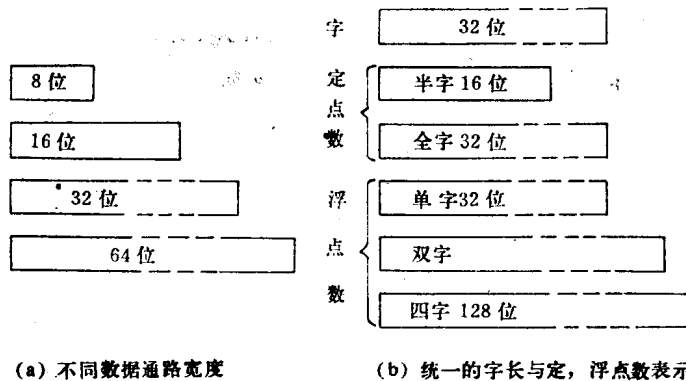


图 1.4 IBM370 系列机的概念性结构

在中央处理机中，它们都执行相同的指令系统，但在低档机上可以采用指令的分析和指令的执行顺序进行的方式，而在高档机上则采用重迭、流水和其它并行处理方式。从程序设计者看，各档机器都具有相同的 32 位字长，即定点数都是统一的半字长或全字长，而浮点数也都是统一的单、双、四倍字长，但从低档到高档机器，其数据通路的宽度可分别具有 8 位、16 位、32 位、甚至 64 位（如图 1.5 所示）。



(a) 不同数据通路宽度

(b) 统一的字长与定、浮点数表示

图 1.5 IBM370 系列机的数据通路和字长

对于VAX—11系列机,DEC公司先后推出VAX—11/780,750,730,725,MICRO—VAX,785等型号。它们都具有32位的字长,都能运行VAX/VMS(虚拟管理系统)的操作系统,都具有相同的I/O联结和使用方式,即I/O设备寄存器和主存统一编址,并统一占用主存的指定单元,程序员通过访问主存指定单元的指令实现对设备读/写操作。但在组成上有的型号机器(如780)采用同步底板互连总线SBI作为CPU、主存子系统、控制台子系统、单总线适配器UBA和多总线适配器MBA的互连部件,如图1.6所示。但VAX—11/730没有多总线,而VAX—11/750甚至没有SBI,也没有控制台子系统,而用集中控制台取而代之。因此,采用哪种总线结构对程序员来讲是透明的,它们是为了适应不同的速度要求而设计的。

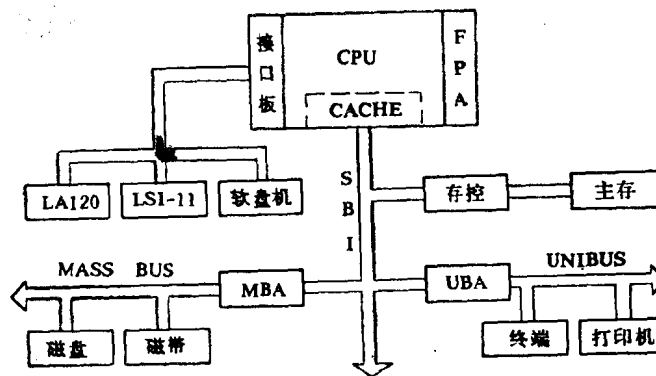


图 1.6 VAX—11/780的结构

从上述系列机的例子可见:一种体系结构可以有多种组成。同样,一种组成可以有多种物理实现。正因为系列机从程序设计者看都具有相同的机器属性,因此按这个属性(体系结构)编制的机器语言程序以及编译程序都能通用于各档机器,我们称这种情况下的各档机器是软件兼容的(Software Compatibility),即同一个软件可以不加修改地运行于体系结构相同的各档机器,而且它们所获得的结果一样,差别只在于有不同的运行时间。长期以来,软件工作者希望有一个稳定的环境,使他们编制出来的程序能得到广泛的应用,机器设计者又希望根据硬件技术和器件技术的进展不断地推出新的机器,而系列机的出现较好地解决了软件要求环境稳定和硬件、器件技术迅速发展之间的矛盾,对计算机的发展起到了重要的推动作用。有些计算机厂家为了能利用大的计算机厂家在软件方面的开发成果,也研制一些软件兼容的产品。我们把不同厂家生产的具有相同体系结构的计算机称为兼容机(Compatible Machine)。兼容机一方面由于采用新的计算机组成和实现技术,因此具有较高的性能价格比;另一方面又可能对原有的体系结构进行某种扩充,使它具有更强的功能(如长城0520微机兼容IBM PC,但有较强的汉字处理功能)。因此,在市场上有较强的竞争能力。更有甚者,以Amdahl公司为代表的接插兼容机PCM(Plug-Compatible Mainframe)厂家,专门生产在功能上和电气性能上与IBM公司相同的主机和设备,它不但可以运行IBM公司的软件,而且又可以作为IBM产品的替换件插入IBM系统。由于采用了新的硬件和器件技术,也改善了性能

价格比，因而成为IBM公司的强有力的竞争对手，迫使IBM公司不断地推出具有新的体系结构的系列机，以削弱兼容机厂家的竞争能力。

系列机为了保证软件的兼容，要求体系结构不准改变，这无疑又成为妨碍计算机体系结构发展的重要因素。实际上，系列机的软件兼容还有向上兼容、向下兼容、向前兼容和向后兼容之分。所谓向上（下）兼容指的是按某档机器编制的程序，不加修改的就能运行于比它高（低）档的机器。所谓向前（后）兼容指的是按某个时期投入市场的某种型号机器编制的程序，不加修改地就能运行于在它之前（后）投入市场的机器。为了适应系列机中性能不断提高或者应用领域不断扩大的要求，后续各档机器的体系结构也是可以改变的。如增加浮点运算指令以提高速度，或者增加事务处理指令以满足事务处理方面的需要等等。但这种改变一定要是原有体系结构的扩充，而不是任意地更改或缩小。这样，对系列机的软件向下和向前兼容可以不作要求，向上兼容在某种情况下也可能做不到（如在低档机器上增加了面向事务处理的指令），但向后兼容却是肯定要做到的。因此，可以说向后兼容是软件兼容的根本特征，也是系列机的根本特征。一个系列机的体系结构设计的好坏，是否有生命力，就看是否能在保证向后兼容的前提下，不断地改进其组成和实现。向后兼容虽然削弱了对体系结构发展的约束，但仍然是体系结构发展的一个沉重的包袱。

对于给定的计算机体系结构，选择什么样的组成和实现技术，应从性能价格比的优化出发。所谓优化，一般是指在当时器件、硬件技术条件下，如何在价格不增或只增很少的前提下，尽可能地提高性能。价格则是不同组成和所用器件集成度、速度、微组装技术、可靠性措施等多种因素的函数。

另外，还要看到：体系结构、组成和实现是相互影响的。一方面体系结构要影响到计算机组成和实现。例如计算：

$$A := B + C$$

$$D := E * F$$

对于面向寄存器的体系结构，其程序可能是：

```

1  LAD  R1, B
2  ADD  R1, C
3  STR  R1, A
4  LAD  R2, E
5  MUP  R2, F
6  STR  R2, D

```

而对于面向主存的具有三地址寻址方式的体系结构，其程序可以是：

```

1  ADD  B, C, A
2  MUP  E, F, D

```

为了提高机器的运算速度，在组成上都采用相加和相乘并行的方式。因此，这两种体系结构都要求分别设置加法器和乘法器。但是，对于面向寄存器的体系结构，还需做到面向R₁的指令和面向R₂的指令能并行执行。而对于面向主存具有三地址寻址方式的体系结构，却要求快速地并行形成操作数地址，且能并行访问主存。由于体系结构影响到计

计算机组成和实现,因此,对于系列机,在确定其体系结构时,既要考虑到它能在低档机上用简单、便宜的组成(如通常的取指令和取操作数之间顺序执行的方法)实现;也要考虑到在高档机上,当采用复杂、昂贵的组成(如流水线方法)时,这种体系结构仍然能够充分发挥该实现方法所带来的好处。

另一方面,计算机组成和实现也会影响到体系结构。如果没有近四十年来计算机组成和实现技术的显著进步,体系结构的进展也是不可能的。特别是微程序技术的实现,能使“软件固化”和“硬件软化”,使软硬件之间不再有明显的界限,引起了体系结构的深刻变化。

由此看出:计算机体系结构一方面决定于应用和软件的要求,以及所采用的算法;另一方面也受组成和实现的影响。因此在设计体系结构时,既要考虑应用的要求,对软件和算法提供更多更好的支持;也要考虑到准备采用和可能采用的组成技术,并且使体系结构尽可能不要对组成、实现技术的采用和发展有过多或不合理的限制。尤其对系列机来讲,这种体系结构才有生命力。

同样,计算机组成的设计一方面决定于计算机体系结构,另一方面受限于所用的实现技术。前者指的是如何把给定的体系结构映象到特定的组成,后者指的是对给定的器件技术,宜于实现什么样复杂程度的组成。而对超大规模集成电路,则是指能将什么样复杂程度的体系结构和组成映象到单片上。

最后,还应看到:体系结构、组成和实现所包含的具体内容是随机器而变化的。有些计算机系统作为体系结构的内容,其它的计算机系统可能作为组成和实现的内容;开始是作为组成和实现提出来的设计思想,到后来就可能被引入到体系结构中去。例如:快速缓冲存贮器的引入一般是属于组成的内容,其中存放的信息是由硬件自动管理的,对程序员来说是透明的。然而,有的机器为了提高缓冲存贮器的效率,使得在更多的情况下都能从缓冲存贮器中取得所需的数据,为此设置了缓冲存贮器的管理指令,使程序员能参与缓冲存贮器内容的确定。在这种情况下,缓冲存贮器又成为体系结构的一部分,对程序员来讲又是不透明的。

体系结构、组成和实现之间的紧密关系在微处理机中表现得更为突出。组成和实现技术的变化将导致具有新的体系结构的微处理机芯片的出现。

既然计算机体系结构设计任务是进行软、硬件功能分配,是传统机器级的软、硬件界面的确定,因此一个好的体系结构设计者应对计算机系统的软、硬件技术都要有较深入的了解和掌握。本书围绕着体系结构和组成这两方面的内容而编写,研究计算机系统中的软、硬件功能的分配,以及如何最佳、最合理地实现分配给硬件的功能。

注意:有的文献把计算机体系结构称为从程序设计者看到的计算机体系结构,而把计算机组成称为从计算机设计者看到的计算机体系结构。

1.3 计算机体系结构和并行性概念的发展

如上所述,计算机体系结构的实质是指计算机系统中软、硬件的界面,即研究软硬件的功能分配,哪些功能由软件完成,哪些功能由硬件完成。实际上,软件和硬件在逻辑