

计算机系统结构

郑纬民 陈修环 编
石 砦 汤志忠

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书讲述计算机系统结构的基本概念、基本原理、基本结构和分析方法,比较系统地讨论了高性能计算机系统的各种技术。

全书共分八章。其内容分别是:绪论;存储器系统设计;流水线计算机设计技术;数值应用问题的特征;向量计算机;多处理机;多处理机算法;数据流计算机系统结构。

本书适合于作计算机专业本科生和有关专业研究生的教材,也可作为计算机科学工作者(包括软件工作者)的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社
激光防伪标志,无标志者不得销售。

计 算 机 系 统 结 构

郑纬民等 编

责任编辑 贾仲良

☆

清华大学出版社出版

(北京 清华园)

通县大中印刷厂印刷

新华书店总店科技发行所发行

开本: 787×1092 1/16 印张: 16 字数: 380 千字

1992年2月第1版 1998年1月第7次印刷

印数: 36001~41000

ISBN 7-302-00954-6/TP·350

定价: 13.80 元

前 言

本书是“计算机系统结构”课程的教材,它适于作为计算机专业本科生和有关专业研究生的教材,也可作为计算机科学工作者的参考书。

本书力求深入浅出地阐述计算机系统结构的基本概念、基本原理、基本结构和基本分析方法。介绍了近几年来出现的高性能计算机的新概念和新技术。重点讨论了存储器设计、流水线技术和各种不同的并行结构。在讨论各种系统结构时紧紧围绕系统可能出现的瓶颈和克服瓶颈的方法。例如,存储器带宽、处理机带宽、I/O带宽、通信带宽和同步技术等。本书力求反映现代计算机发展的水平。各章的内容互相独立,教师可以根据不同的专业、不同的学时任意选择几章进行教学。

本书第一、四、五、六、七章由郑纬民编写,第二章由陈修环编写,第三章由石畧编写,第八章由汤志忠编写。

本书的出版得到了清华大学计算机科学与技术系领导和老师们的支持和鼓励,在此一并致谢。

限于作者的经验与水平,错误和不当之处希望读者批评指正。

作者

1990年12月

目 录

第一章 绪论	1
1.1 计算机系统结构	1
1.1.1 计算机系统层次结构.....	1
1.1.2 计算机系统结构定义.....	3
1.1.3 计算机组成与实现.....	3
1.1.4 计算机系统结构的发展.....	4
1.1.5 计算机系统结构的分类.....	5
1.2 技术和系统结构	8
1.3 系统结构的评价标准	9
1.3.1 成本指标.....	9
1.3.2 硬件考虑	10
1.4 高性能技术.....	12
1.4.1 价格对系统结构的影响	12
1.4.2 应用对系统结构的影响	13
1.4.3 VLSI 对系统结构的影响	14
1.4.4 技术的发展对价格的影响	14
1.4.5 算法和系统结构	16
第二章 存储系统设计	18
2.1 利用程序本身的特性.....	19
2.2 Cache 存储器	22
2.2.1 Cache 基本结构	22
2.2.2 Cache 设计	24
2.2.3 Cache 分析	28
2.2.4 Cache 替换算法	39
2.2.5 在 Cache 中的进程足迹	43
2.2.6 Cache 写操作	48
2.3 虚拟存储器.....	50
2.3.1 虚拟存储器结构	51
2.3.2 虚拟存储器的地址变换	53
2.3.3 分段存储系统	57
2.3.4 替换算法	58
2.3.5 缓冲对虚拟存储系统性能的影响	62
第三章 流水线计算机设计技术	66

3.1	流水线设计原理	66
3.2	流水线计算机中的存储器结构	73
3.3	流水线计算机的性能	74
3.4	流水线中段的控制方法	78
3.4.1	多功能流水线的设计	78
3.4.2	冲突向量和流水线控制	80
3.4.3	流水线的最佳性能	83
3.4.4	利用延迟线改善性能	85
3.4.5	消除互锁	89
3.5	流水线技术的开发	90
3.5.1	条件转移	90
3.5.2	内部定向和推迟指令	93
3.5.3	具有 Cache 存储器和虚拟存储器的机器	98
3.5.4	RISC 结构	100
第四章	数值应用问题的特征	103
4.1	大型数值问题的分类	103
4.1.1	连续模型	104
4.1.2	粒子模型	105
4.2	高性能计算机设计中的限制条件	107
4.3	连续模型的系统结构	108
4.4	连续模型的算法	112
4.4.1	宇宙立方体结构	113
4.4.2	数据流需求	113
4.4.3	并行解法	116
4.4.4	递归加倍法与循环归约法	119
4.5	混洗	122
4.5.1	混洗互连模式	122
4.5.2	全混洗的应用	125
4.6	连续模型的结构向何处发展	130
第五章	向量计算机	132
5.1	向量处理的基本概念	132
5.1.1	什么是向量处理	132
5.1.2	向量处理方式	133
5.2	一般的向量处理机	134
5.2.1	多存储器模块	136
5.2.2	中间存储器	140
5.3	数值算法的存取模式	143
5.3.1	高斯消去法	143

5.4	向量机的数据结构	145
5.5	向量协处理器	149
5.6	稀疏矩阵的处理技术	152
5.7	一台超高速的向量处理机——GF-11	153
5.8	关于向量计算机的几点看法	154
第六章	多处理机	157
6.1	背景	157
6.2	多处理机性能	159
6.2.1	基本模型	161
6.2.2	N台处理机系统的基本模型	162
6.2.3	随机模型	164
6.2.4	通信开销为线性函数的模型	164
6.2.5	一个完全重叠通信的理想模型	165
6.2.6	一个具有多条通信链的模型	167
6.2.7	多处理机模型	168
6.3	多处理机的互连网络	169
6.3.1	总线互连	170
6.3.2	环形互连	172
6.3.3	交叉开关互连	173
6.3.4	混洗交换互连和合并开关	177
6.3.5	蝶形操作	178
6.3.6	合并网络和取与加指令	181
第七章	多处理机算法	185
7.1	简单并行性	185
7.1.1	do par 和 do seq 结构	187
7.1.2	阻塞同步	187
7.1.3	性能分析	188
7.1.4	增大粒度	190
7.1.5	任务的初始化	192
7.2	同步技术	194
7.2.1	使用测试与设置指令的同步技术	195
7.2.2	使用增1和减1指令的同步技术	197
7.2.3	使用比较与交换指令的同步技术	199
7.2.4	使用取与加指令的同步技术	203
7.3	并行搜索	205
7.3.1	搜索单峰函数的极大值	205
7.3.2	并行分支限界法	207
7.4	串行算法到并行算法的转换	210

7.4.1	相关性分析	211
7.4.2	开发迭代的并行性	212
7.5	同步并行算法和异步并行算法	214
7.5.1	同步并行算法	214
7.5.2	异步并行算法	216
7.6	小结	219
第八章	数据流计算机系统结构	221
8.1	数据流驱动原理	221
8.1.1	串行控制流与并行控制流	221
8.1.2	数据流计算机中指令的执行过程	222
8.1.3	数据流计算机的指令组成	223
8.1.4	数据流计算机模型	224
8.2	数据流计算机的性能分析	226
8.2.1	数据流计算机的优点	226
8.2.2	数据流计算机的缺点	227
8.2.3	数据流计算机设计中需要解决的几个主要问题	228
8.3	数据流程序图和数据流语言	228
8.3.1	数据流程序图	229
8.3.2	数据流语言	233
8.3.3	数据流语言的性质	234
8.4	数据流计算机结构	235
8.4.1	静态数据流计算机结构	236
8.4.2	动态数据流计算机结构	240
8.4.3	其它类型的数据流计算机	244

第一章 绪 论

自第一台电子计算机问世已经四十多年,它经历了电子管、晶体管、集成电路和大规模集成电路等四代。目前世界各国正在研制第五代计算机。换代的标志主要有两个:第一是计算机的器件。器件发生了根本的变化,经电子管、晶体管发展到集成电路,而集成电路又由小规模、中规模,到大规模和超大规模的阶段。器件的更新,其速度、功能、可靠性的不断提高和成本的不断降低,是计算机发展的物质基础。因此,器件的换代是计算机换代的最突出的标志。第二是系统结构的特点。系统结构不断改进,许多重要概念不断提出并且得到实现。例如变址寄存器概念,通用寄存器概念、浮点数据表示、程序中断概念、输入输出通道概念、间接寻址概念、虚拟存储器概念、Cache 存储器概念、系列化概念、微程序设计技术等。很明显,如果用大规模集成电路实现早期的计算机系统结构,人们并不会认为它是第四代计算机。因为第四代计算机在系统结构上较之早期的计算机已经有很大的改进和发展。因此,系统结构方面的特点同样是计算机换代的重要标志。

回顾计算机的发展历史,可以看出,计算机系统性能的不断f提高主要靠器件的变革和系统结构的改进。恩斯洛(P. H. Enslow)曾经比较了 1965~1975 这十年间,器件延迟时间和计算机指令时间的关系。结果表明,这十年间器件延迟时间降低至原来的十分之一,但计算机的指令时间却以高至 100 倍的速度递减。这种情况在计算机近年来的发展中变得更加明显。如何最合理地利用新器件,最大限度地发挥其潜力,设计并构成综合性能指标最佳的计算机系统,但是单纯依靠器件变革是不能解决的,还要靠计算机系统结构上的改进。

本书专门研究计算机系统结构,特别是高性能计算机系统结构。重点在于系统结构的设计和分析。

那么,什么是计算机系统结构?计算机系统结构和技术有什么关系?计算机系统结构的评价标准是什么?高性能计算机系统的技术是什么?本章就是要解决这些问题。

1.1 计算机系统结构

1.1.1 计算机系统层次结构

计算机系统由硬件/器件和软件组成,按功能划分成多级层次结构,如图 1.1 所示。图中每一级各对应一种机器,其作用和组成如图 1.2 所示。在这里,“机器”只对一定的观察者而存在。它的功能体现在广义语言上,能对该语言提供解释手段,如同一个解释器,然后作用在信息处理和控制对象上。从某一层次的观察者看来,他只是通过该层次的语言来了解和使用计算机,不必关心再内层的那些机器是如何工作和如何实现各自功能的。

图 1.1 中的第 0 级机器由硬件实现,第 1 级机器由微程序(固件)实现,第 2 级至第 6

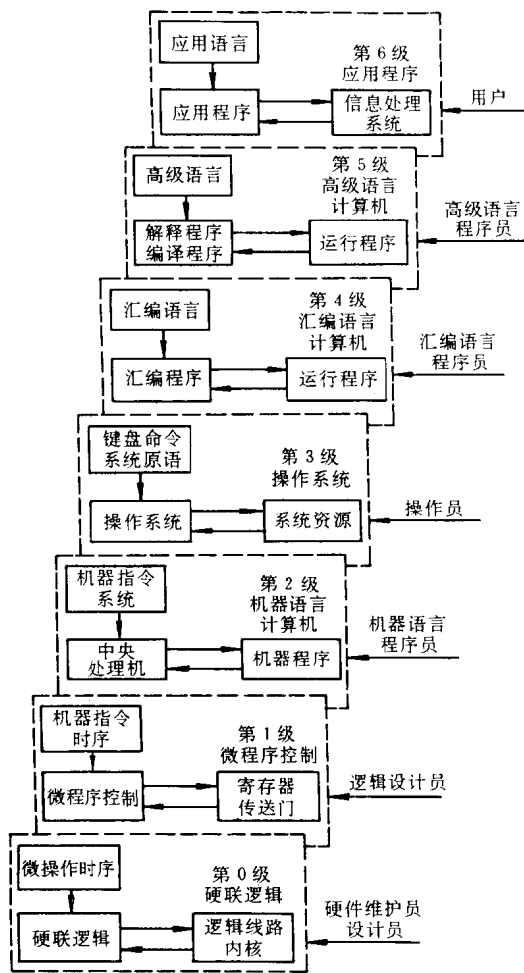


图 1.1 计算机系统层次结构

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。完成翻译的程序叫做汇编程序。

第 5 级是高级语言机器。这级的机器语言就是各种高级语言。用这些语言所编写的程序一般是由编译程序翻译到第 4 级或第 3 级上的语言,个别的高级语言也用解释的方法实现。

第 6 级是应用语言机器。这级的机器语言是应用语言。这种语言使非计算机专业人员也能直接使用计算机,只需在用户终端用键盘或其它方式发出服务请求就能进入第 6 级的信息处理系统。

级机器由软件实现。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。

第 0 级和第 1 级是具体实现机器指定功能的中央控制部分。它根据各种指令操作所需要的控制时序,配备一套微指令,编写出微程序,控制信息在各寄存器之间的传送,这就是第 1 级机器。实现这些微指令本身的控制时序只需要很少的逻辑线路,可采用硬联逻辑实现,它就是第 0 级机器,是机器的硬件内核。

第 2 级是传统机器语言机器。这级的机器语言是该机的指令系统。机器语言程序员用这级指令系统编写的程序由第 1 级的微程序进行解释。

第 3 级是操作系统机器。这级的机器语言中的多数指令是传统机器的指令,如算术运算,逻辑运算和移位等指令。此外,这一级还提供操作系统级指令,例如打开文件、读/写文件、关闭文件等指令。用这一级语言编写的程序,即那些与第 2 级指令相同的指令直接由微程序实现。操作系统级指令部分由操作系统进行解释。操作系统是运行在第 2 级上的解释程序。

第 4 级是汇编语言机器。这级的

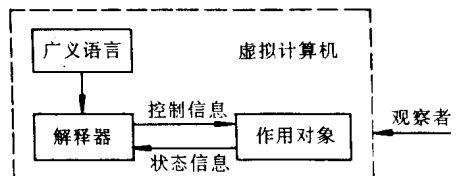


图 1.2 机器的作用和组成

从学科领域来划分,大致可以认为第0至第1级是计算机组织与结构讨论的范围,第3至第5级是系统软件,第6级是应用软件。但是,严格说起来又不尽然,它们之间仍有交叉。例如,第0级要求一定的数字逻辑基础;第2级涉及汇编语言程序设计的内容;第3级与计算机系统结构密切相关。在特殊的计算机系统中,有些级别可能不存在。

把计算机系统按功能划分成多级层次结构,首先有利于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 and 作用。其次有利于理解各种语言的实质及其实现。最后还有利于探索虚拟机新的实现方法,设计新的计算机系统。

1.1.2 计算机系统结构定义

计算机系统结构这个词是 Amdahl 等人在 1964 年提出的。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。按照计算机层次结构,不同程序设计者所看到的计算机有不同的属性。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。Amdahl 等人提出的系统结构定义中的程序设计者是指为机器语言或编译程序设计者所看到的计算机属性,是硬件子系统的概念结构及其功能特性,包括机器内的数据表示,即硬件能直接辨认和处理的那些数据类型;寻址方式,包括最小寻址单元和地址运算等;寄存器定义,包括操作数寄存器、变址寄存器、控制寄存器等的定义、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,包括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;输入-输出结构,包括输入-输出的连结方式,处理机/存储器与输入-输出设备间数据传送的方式和格式、传送的数据量、以及输入-输出操作的结束与出错标志等;信息保护,包括信息保护方式和硬件对信息保护的支持等等。这些即是程序员为了使其所编写的程序能在机器上正确运行,需要了解和遵循的计算机属性。当然不包括基本的数据流、控制流、逻辑设计和物理实现等。

计算机系统结构作为一门学科,主要研究软件、硬件功能分配和对软件、硬件界面的确定,即哪些功能由软件完成,哪些功能由硬件完成。

关于计算机系统结构这一概念,至今有各种各样的理解,很难有一个通用的定义。在下节讨论计算机组成和实现后,我们还要给出另一些定义。

1.1.3 计算机组成与实现

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大、速度快,人们研究出层次存储系统和虚拟存储技术。在层次存储系统中,又有高速缓存、多模块交错工作、多寄存器组和堆栈等技术。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,人们研究出先行控制、流水线、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。例如,在运算器方面,出现了多种自动调度算法和结构等。

由此可见,计算机组成是计算机系统结构的逻辑实现,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,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系统结构。一般计算机组成设计包括数据通路宽度的确定,各种操作对功能部件的共享程度的确定,专用功能部件的确定,功能部件的并行性确定,缓冲器和排队的确定,控制机构的设计,可靠性技术的确定等。

计算机实现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。

计算机系统结构、计算机组成和计算机实现是三个不同的概念。系统结构是计算机系统的软、硬件的界面;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,但又有紧密的关系。

我们还应看到系统结构、组成和实现所包含的具体内容是随不同机器而变化的。有些计算机系统是作为系统结构的内容,其它计算机系统可能是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系统结构中。例如高速缓冲存储器一般是作为组成提出来的,其中存放的信息全部由硬件自动管理,对程序员来说是透明的。然而,有的机器为了提高其使用效率,设置了高速缓冲存储器的管理指令,使程序员能参与高速缓冲存储器的管理。这样,高速缓冲存储器又成为系统结构的一部分,对程序员来说是不透明的。

Amdahl 等人的计算机系统结构定义的主要内容是指令系统及其执行模型。根据这个定义,一个系列机中不同档次的机器有相同的系统结构。Amdahl 等人定义系统结构时认为只要指令系统兼容就能保证程序正确运行。由于程序的执行要依赖于程序库、操作系统和其它 Amdahl 等人的系统结构定义中没有涉及的因素,这要求操作系统接口等其它层次标准化。同时,由于 VLSI 的迅速发展及其成本急剧下降,有些系列机推出有新指令的机器,例如 24 位地址的 IBM 360 和 370 系统发展为 31 位地址的 370xA 系统,16 位地址的 PDP-11 发展为 32 位地址的 VAX 系列。随着新器件的出现,当今计算机设计者面临的问题与十年前面临的问题大不相同,所以我们应当把计算机系统结构定义得更宽一些,除了 Amdahl 等人定义的内容外,还应包括功能模块的设计。也就是说,计算机系统结构、计算机组成、计算机实现之间的界限越来越模糊了。

1.1.4 计算机系统结构的发展

冯·诺依曼等人于 1946 年提出了一个完整的现代计算机雏型,它由运算器、控制器、存储器和输入-输出设备组成,如图 1.3 所示。

现代的计算机系统结构与冯·诺依曼等人当时提出的计算机系统结构相比虽已发生了重大变化,但就其结构原理来说占有主流地位的仍是以存储程序原理为基础的冯·诺依曼型计算机。存储程序原理的基本点是指令驱动,即程序由指令组成,并和数据一起存

放在计算机存储器中,机器一经启动,就能按照程序指定的逻辑顺序把指令从存储器中读出来逐条执行,自动完成由程序所描述的处理工作。冯·诺依曼计算机的特征可概括为:

1. 存储器是字长固定的、顺序线性编址的一维结构。

2. 存储器提供可按地址访问的一级地址空间,每个地址是唯一定义的。

3. 由指令形式的低级机器语言驱动。

4. 指令的执行是顺序的,即一般按照指令在存储器中存放的顺序执行,程序分支由转移指令实现。

5. 机器以运算器为中心,输入-输出设备与存储器之间的数据传送都途经运算器。运算器、存储器、输入-输出设备的操作以及它们之间的联系都由控制器集中控制。

虽然至今绝大多数计算机仍基于上述结构特点,但这四十多年来计算机系统结构有了许多改进。主要包括以下几个方面:

1. 计算机系统结构从基于串行算法改变为适应并行算法,从而出现了向量计算机,并行计算机、多处理机等。

2. 高级语言与机器语言的语义距离缩小,从而出现了面向高级语言机器和直接执行高级语言机器。

3. 硬件子系统与操作系统和数据库管理系统软件相适应,从而出现了面向操作系统机器和数据库计算机等。

4. 计算机系统结构从传统的指令驱动型改变为数据驱动型和需求驱动型,从而出现了数据流机器和归约机。

5. 为了适应特定应用环境而出现了各种专用计算机,如快速傅里叶变换机器、过程控制计算机等。

6. 为了获得高可靠性而研制容错计算机。

7. 计算机系统功能分散化、专业化,从而出现了各种功能分布计算机,这类计算机包含外围处理机、通信处理机等。

8. 出现了与大规模、超大规模集成电路相适应的计算机系统结构。

9. 出现了处理非数值化信息的智能计算机。例如自然语言、声音、图形和图象处理等。主要的处理方法已不是依靠精确的算法进行数值计算而是依靠有关的知识进行逻辑推理,特别是利用经验性知识对不完全确定的事实进行非精确性推理。

1.1.5 计算机系统结构的分类

下面介绍三种不同的分类方法。

1. Flynn 分类法

1966年 M. J. Flynn 提出了如下定义:

指令流 (Instruction Stream)——机器执行的指令序列。

数据流 (Data Stream)——由指令流调用的数据序列,包括输入数据和中间结果。

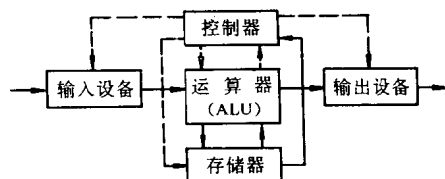


图 1.3 早期的冯·诺依曼型机器组成框图

多倍性(Multiplicity)——在系统最受限制的元件上同时处于同一执行阶段的指令或数据的最大可能个数。

同时,他按照指令流和数据流的不同组织方式,把计算机系统的结构分为以下四类:

- (1) 单指令流单数据流 SISD(Single Instruction Stream Single Data stream)
- (2) 单指令流多数据流 SIMD(Single Instruction stream Multiple Data stream)
- (3) 多指令流单数据流 MISD(Multiple Instruction stream Single Data stream)
- (4) 多指令流多数据流 MIMD(Multiple Instruction stream Multiple Data stream)

对应于这四类计算机的基本结构框图如图 1.4 所示。SISD 是传统的顺序处理计算机。SIMD 以阵列处理机或并行处理机为代表。MISD 在实际上代表何种计算机,存在着不同的看法,有的文献把流水线结构机器看作是 MISD 结构。多处理机属于 MIMD 结构。

2. 冯氏分类法

冯泽云于 1972 年提出用最大并行度对计算机系统结构进行分类。最大并行度 P_m 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。假定每个时钟周期 Δt_i 内能同时处理的二进制位数为 P_i ,则 T 个时钟周期内平均并行度为

$$P_a = \frac{\sum_{i=1}^T P_i \Delta t_i}{T}$$

平均并行度不同于最大并行度,它取决于系统的运用程度,与应用程序有关。因此,定义系统在周期 T 内的

平均利用率为
$$\mu = \frac{P_a}{P_m} = \frac{\sum_{i=1}^T P_i}{TP_m}$$

图 1.5 示出用最大并行度对计算机系统结构进行分类的方法。用平面直角坐标系中的一点代表一个计算机系统,横坐标代表字宽(n 位),即在一个字中同时处理的二进制位数;纵坐标代表位片宽度(m 位),即在一个位片中能同时处理的字数。于是,一个系统的最大并

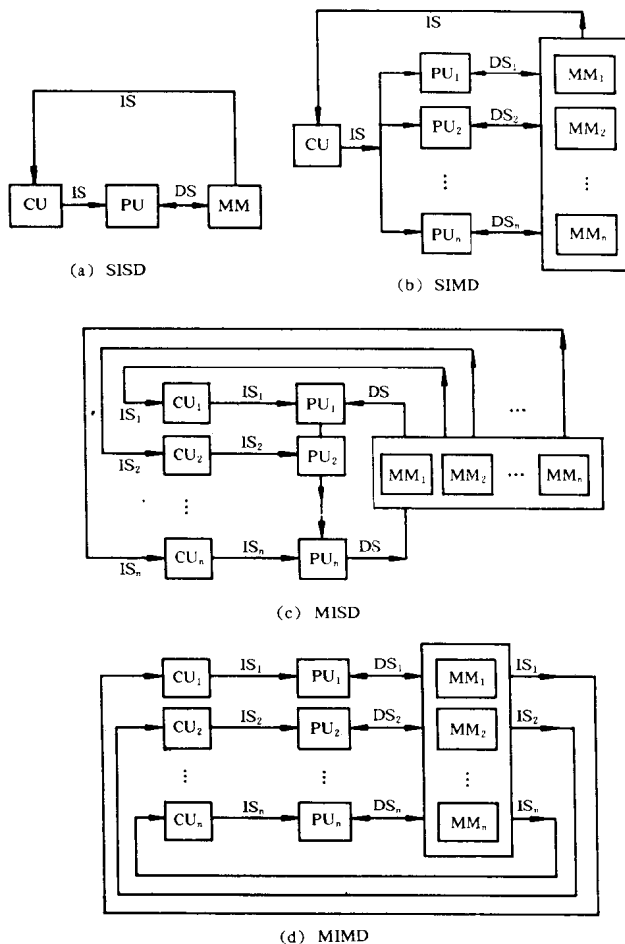


图 1.4 Flynn 分类法各类机器结构

行度就可以用这两个量的乘积,即用通过该点的水平线和垂直线与两坐标轴围成的矩形面积来表示。

由图 1.5 可得出四类不同处理方法的计算机系统结构：

(1) 字串位串 WSBS (Word-Serial and Bit-Serial), 其 $n=1, m=1$ 。这是第一代计算机发展初期的纯串行计算机。

(2) 字并位串 WPBS (Word-Parallel and Bit-Serial), 其 $n>1, m=1$ 。这是传统并行单处理机。

(3) 字串位并 WSBP (Word-Serial and Bit-Parallel), 其 $n=1, m>1$ 。STARAN, MPP, DAP 属于这种结构。

(4) 字并位并 WPBP (Word-Parallel and Bit-Parallel), 其 $n>1, m>1$ 。PEPE, ILLIAC IV, Cmpmp 属于这种结构。

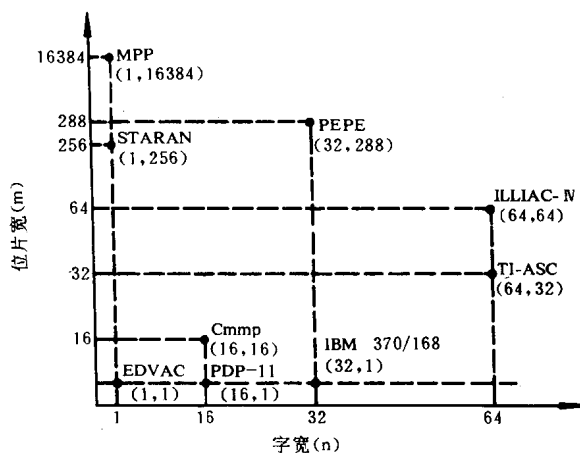


图 1.5 按最大并行度的冯氏分类法

3. Handler 分类法

Wolfgang Handler 在 1977 年根据并行度和流水线提出了另一个分类法。这种分类方法把计算机的硬件结构分成三个层次,并分别考虑它们的可并行—流水处理程度。这三个层次是：

- (1) 程序控制部件(PCU)的个数 K ;
- (2) 算术逻辑部件(ALU)或处理部件(PE)的个数 d ;
- (3) 每个算术逻辑部件包含基本逻辑线路(ELC)的套数 w 。

这样我们可以把一个计算机系统的结构用如下公式表示：

$$t(\text{系统型号}) = (k, d, w)$$

为了进一步揭示流水线的特殊性,一个计算机系统的结构可用如下公式表示：

$$t(\text{系统型号}) = (k \times k', d \times d', w \times w')$$

式中: k' 表示宏流水线中程序控制部件的个数。 d' 表示指令流水线中算术逻辑部件的个数。 w' 表示操作流水线中基本逻辑线路的套数。

例如 Cray 1 有 1 个 CPU, 12 个相当于 ALU 或 PE 的处理部件, 可以最多实现 8 级流水线。字长为 64 位, 可以实现 1~14 位流水线处理。所以 Cray 1 的系统结构可表示为：

$$t(\text{Cray 1}) = (1, 12 \times 8, 64(1 \sim 14))$$

下面是用这种分类法的例子：

$$t(\text{PDP 11}) = (1, 1, 16)$$

$$t(\text{ILLIAC IV}) = (1, 64, 64)$$

$$t(\text{STARAN}) = (1, 8192, 1)$$

$$t(\text{Cmpmp}) = (16, 1, 16)$$

$$t(\text{PEPE}) = (1 \times 3, 288, 32)$$

$$t(\text{TIASC}) = (1, 4, 64 \times 8)$$

1.2 技术和系统结构

技术的发展促进了计算机系统结构的发展。每年都有新的器件和新的设备问世,使新功能的计算机系统结构不断出现。今天看来富有想象力和卓有成效的系统结构,明天也许会显得落后了。同样,今天看来是荒谬、不切实际的建议,明天可能是很理想的设计方案。因此不存在一个绝对的标准用来断言一种系统结构一定比另一种系统结构好。

学习计算机系统结构的关键是学会在现有技术条件下评价系统结构的方法。了解一台计算机的速度固然重要,但了解其是否充分利用了处理机周期、存储器容量、输入-输出带宽等也同样重要。评价一个系统结构既要考虑价格又要考虑性能,不能只注意性能。由于技术的发展,各功能模块的价格每年都有很大的变化,各功能模块之间的相对价格同样也有很大的变化,所以为了获得较好的性能/价格比,不同功能模块的最佳比例也应随着技术的发展而有所改变。

本书将介绍系统结构的设计技术,而不是仅仅给出结论。我们准备提供一系列可供选择的设计技术,其中有些对今天来说是合理的,有些则不合理。我们将介绍如何从中选择合适的技术来构造高性能的系统和如何评价采用 80 年代技术所制造的系统。评价所获得的结论一直到 90 年代可能都是合理的,但是我们无论如何也不能说今天看来是最佳的系统结构在下一个十年仍是最佳的。

然而,方法是长期有效的。系统结构设计者使用本书所介绍的设计方法与评价技术就可以在任何时候根据当时的技术,设计出高性能的系统。

性能分析应建立在整个系统结构基础上。设计和分析一个高性能系统非常复杂,最好的方法是把一个系统分解成由功能模块组成的层次形式,这样每个功能模块的系统结构可以分别进行分析。如果某个功能模块仍然非常复杂,那么还可以进一步分解成更小的功能模块。例如,处理机可以看作由算术部件、控制逻辑、寄存器等组成。

指令系统的设计是系统结构的一个重要内容。目前存在着指令系统是非常复杂好还是非常简单好的争论。我们不打算介入这场争论,因为关于这个争论不可能只有一个答案。但是我们要阐明决定这个答案的各种因素。不管采用什么技术,系统结构设计者在进行新的设计时应应对这些因素进行权衡。

往往易把计算机系统结构设计和计算机硬件设计相混淆。由于计算机系统结构的设计是在功能这一层次上考虑问题,当然也不排斥硬件这一层次,但计算机系统结构设计不只包括硬件设计。例如,一台具有算术和逻辑功能的处理器可以由硬件实现而无需额外的程序设计。存储器管理功能可以由硬件和软件共同实现,它们之间的分工取决于性能、价格及当前硬件和软件的可用性情况。在 VLSI 还处于发展初期时,存储器管理功能一般由软件实现,处理机必须提供用于地址变换和地址保护的寄存器。随着 VLSI 的不断发展,存储器管理的大部分功能变为由硬件实现。许多系统已经用硬件来实现过去只能用软件方法实现的存储器管理算法。如果把存储器管理软件写到只读存储器,管理程序可以透明地调用它,那么硬件和软件之间的界线就变得模糊了。我们可把这种只读存储器视为一个实现存储器管理的黑匣子。这样,实现存储器管理功能就由软件方法变为硬件方法了。使用

这种只读存储器芯片的系统结构设计者不需要考虑存储器管理软件。如果这种芯片只实现大部分而不是全部的存储器管理功能,那么系统结构设计者还必须通过软件模块来弥补不足的功能。

综上所述,计算机系统结构是把各个功能部件组成一个系统,这些部件可以是硬件、软件或两者的混合体。系统结构设计是选择一种最佳的部件组合,使得整个系统能有效地工作。以后章节我们将介绍不同系统结构的实例,其中有些正被证明是成功的,有些则是可能会成功的方案。

1.3 系统结构的评价标准

评价一个计算机系统结构好坏的标准是什么?我们可以用速度、程序和数据的容量、功耗、体积、编程的难易程度、成本等指标来评价。其中最重要的是性能和成本这两个指标。

1.3.1 成本指标

由于许多人对成本是什么含意还不大清楚,所以有必要作一点解释。对用户来说,计算机系统的成本是指购买系统所要付的钱,即价格。对设计者来说,成本的定义就没有那样清楚了。大多数情况下,成本是指生产成本,其中包括开发工具的折旧费用。

在计算机早期年代,软件是随出售的硬件免费赠送的,但随着计算机工业的不断发展,软件本身变成了很有价值的商品。昔日一度作为免费赠送的软件现已成为在计算机系统的预算中占很大部分的产品。软件和硬件的成本变化趋势如图 1.6 所示。从图中我们可以看到,软件的成本随着其复杂性和长度的增加而不断提高,并没有因为软件工具的不断改进而得到明显的缓解。图 1.6 中黑方块表示在同一时期硬件成本的大致趋势。硬件的成本以不可思议的速度下降。如果软件和硬件的成本还是以这种趋势发展下去,那么 10 年到 20 年后,硬件就可能成为在其上运行的软件的附属品而免费赠送了。当然,这种观点现在看来是相当可笑的。

软件和硬件的成本各由两部分组成:

1. 一次性开发成本。
2. 每个部件的生产成本。

不管是硬件还是软件,每种产品的实际成本是该产品产量的函数,如图 1.7 所示。图中曲线是全部产品的累加成本。我们从图中可以看到第一个产品的成本等于开发成本。成本曲线随着产量的增加不断上升。但产量不断增加时,成本曲线上升的趋势变慢,其原因是随着产量的增加,生产经验不断丰富,每个产品的生产成本下降了。产品的单价等于图中曲线值除以产品的数量再

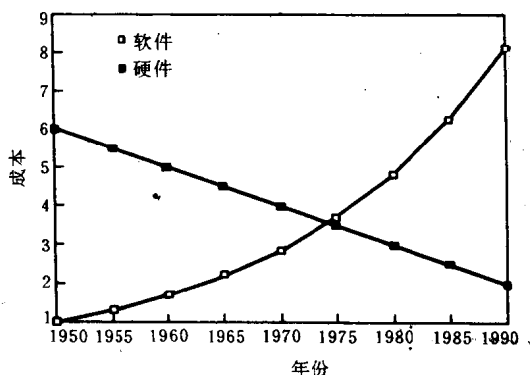


图 1.6 硬件成本和软件成本的变化趋势