

多微处理器系统设计 及其实例

● 侯伯亨 编著



什么是多微处理器系统

西安电子科技大学出版社

前 言

随着计算机技术的发展,多处理机(多微处理器)系统已在许多领域得到了广泛的应用。许多工程技术人员在微机应用系统的设计中也自觉不自觉地用到了多处理机系统设计的相关技术。但是,对于大多数从事计算机应用的科技人员来说,“多处理机系统”这个概念如同“阳春白雪”,高不可攀。究其原因大约是,从80年代以来,国内外都陆续出版或刊出了许多有关“多处理机系统”方面的专著和论文。它们多在理论上对“多处理机系统”作了较大篇幅的分析和论述,而出于种种原因,对多机系统的具体设计却涉及甚少,从而使读者感到它高深莫测。这实在是一种极大的误解。

多处理机系统设计的相关技术和基本知识理应是广大微机应用系统设计人员的宝贵财富,如能被大多数设计人员所掌握,必将使他们所设计的微机应用系统的性能价格比有较大幅度的提高,从而使微机应用的水平上一个新的台阶。普及多机系统的知识是当前技术发展的需要,也是当务之急。作者编写本书的目的也正在于此。

本书是作者在对多年来的教学内容作了适当增删后形成的。它突出了多处理机系统的基本知识的介绍和工程设计实例,使本书更具可读性和实际使用参考价值。在编写过程中力求简明扼要,深入浅出,使得它既能为具有一定微机技术基础的工程技术人员所使用,也可作为大专院校的教师、学生的教科书或参考书。

本书共分9章。第一章对多机系统的一般概念作了综合性介绍。第二章和第三章分别介绍了松耦合多机系统和紧耦合多机系统的互连技术和相应的工程设计问题。第四章介绍了多机系统中的高速缓存技术和实现Cache一致性协议的硬件和软件方法。第五章介绍了多机系统的进程通信和同步。第六章论述了多机系统调度的一般方法,并列举了静态调度和动态调度的实例。第七章以LISP计算机中所采用的负载平衡方法为例,叙述了使用硬件和软件的两种具体实现途径。第八章简介了系统的死锁与预防。第九章介绍了4个多微处理器系统设计的实例。

本书在编写过程中引用了诸多学者、专家的著作和论文中的研究成果,在这里向他们表示衷心的感谢。同时,也向一贯热情支持和关心作者的西安电子科技大学出版社的领导及有关编辑及工作人员表示深深的谢意。

由于作者水平有限,错误和不当之处在所难免,敬请各位读者不吝赐教。

编者

1995年4月11日于西安

目 录

第一章 概述	1
§1-1 计算机技术的发展趋势	1
一、多处理机系统是科学发展的必然趋势	1
二、并行处理与多处理机系统	2
§1-2 多处理机系统的一些基本概念	4
一、多处理机系统的定义和主要特征	4
二、并行处理计算机和多处理机系统的分类	6
三、多处理机系统的发展史	12
四、测试多机系统性能的几个主要方法	15
五、多处理机系统的主要优点	15
§1-3 多机系统应解决的一些主要问题	17
一、程序的分割	17
二、任务的调度	19
三、并行性的检出	20
四、处理机之间的同步处理	20
五、通信开销	21
六、共享资源的存取	21
习题与思考题	22
第二章 松耦合多处理机系统结构及其互连技术	23
§2-1 概述	23
一、松耦合多机系统的特点	23
二、松耦合多机系统的性能评价	23
三、松耦合多机系统的分类	26
§2-2 环形结构	31
一、环路结构的主要特点	31
二、3种单向环结构的基本工作原理	31
三、环路传送介质和环路接口	35
四、环路故障的处理	36
§2-3 公共总线结构	37
一、公共总线结构的特点	37
二、两种单总线结构系统的基本工作原理	38
三、总线传送介质及接口	40
四、总线故障的处理	41
习题与思考题	42

第三章 紧耦合多处理机系统结构及其互连技术	43
§ 3-1 概述	43
一、紧耦合多机系统的特点	43
二、紧耦合多机系统的性能评价	44
三、紧耦合多机系统的分类	44
§ 3-2 公共总线共享存储器互连结构	47
一、系统基本结构	47
二、系统的主要性能指标	48
三、支持多处理器的系统总线(MULTIBUS)	49
四、总线切换逻辑	54
五、总线控制请求的方法	55
§ 3-3 交叉开关共享存储器互连结构	57
一、系统基本结构	57
二、系统的主要性能指标	57
三、交叉开关结点的控制逻辑	58
§ 3-4 多端口共享存储器互连结构	61
一、多端口存储器的结构	61
二、双端口共享存储器设计实例	62
三、集成双端口存储器	63
§ 3-5 共享存储器减少竞争的硬件解决办法	64
一、多读单写存储器(MRSW)	64
二、多读组合存储器(MRM)	64
三、透明存储器(TM)	65
习题与思考题	66
第四章 多机系统中的高速缓存技术	68
§ 4-1 概述	68
一、高速缓存(Cache)的功能	68
二、常用名词解释	69
三、Cache 的工作过程	69
四、Cache 在多微处理器系统中的应用	72
§ 4-2 基于硬件的 Cache 一致性协议	73
一、维持 Cache 一致性的两种策略	73
二、写无效监视 Cache 协议	76
三、写刷新监视 Cache 协议	78
四、Cache 的替换策略	80
五、监视 Cache 协议实施中存在的若干问题与解决办法	82
六、目录方案	84
七、Cache 一致性网络结构	91
§ 4-3 基于软件的 Cache 一致性协议	95

一、基于软件的 Cache 一致性策略的考虑	95
二、可入 Cache 变量的标识	95
三、几种软件的 Cache 一致性实施方案	96
习题与思考题	99
第五章 多处理机系统的进程通信和同步	101
§ 5-1 基于共享变量的进程通信	102
一、常用名词说明	102
二、忙等待	102
三、信号灯同步	105
四、条件临界区	107
五、管程	109
§ 5-2 基于消息传递的进程间通信	111
一、通信协议	111
二、寻址方法	112
三、同步方式	114
习题与思考题	114
第六章 多机系统的调度	115
§ 6-1 概述	115
一、调度的含义和概念	115
二、多处理机系统的任务调度方法和原则	115
§ 6-2 系统的任务调度	117
一、静态调度	117
二、动态调度*	120
习题与思考题	121
第七章 多处理机系统的负载平衡	122
§ 7-1 几种负载平衡的方法	122
一、接收者发动的负载平衡	122
二、发送者发动的负载平衡	123
三、混合负载平衡	124
§ 7-2 任务(进程)迁移的策略	124
一、任务迁移的策略	124
二、任务控制块	125
三、表示程序并行性的命令	126
§ 7-3 混合负载平衡控制的模型及实现	127
一、宏观数据流实现模型	127
二、由硬件来构造混合负载平衡器	128
三、由软件来实现混合负载平衡器	131
习题与思考题	133
第八章 系统的死锁与预防	134

§ 8-1 系统死锁的成因及识别	134
一、系统死锁的成因	134
二、死锁状态的判别与分析	135
§ 8-2 系统死锁的预防	136
一、静态预防方法	136
二、动态预防方法	137
习题与思考题	139
第九章 多处理机系统实例	140
§ 9-1 主从式共享存储器连接的多处理器系统	141
一、航海语音记录子系统所提出的要求	141
二、语音记录子系统的系统结构	141
三、系统主要接口电路设计	142
四、系统设计中的几个实际问题	146
§ 9-2 UNIP 多处理器系统	147
一、UNIP 系统的基本结构	148
二、UNIP 多机系统的软件配置	152
三、UNIP 多机系统的典型应用	157
§ 9-3 多 DSP 数字信号处理系统	164
一、开发多 DSP 数字信号处理系统存在的困难	165
二、4 个 DSP 多处理机系统的结构	165
三、先入为主的无顺序裁决	167
四、预先安排共享存储器请求的裁决	168
五、阻塞同步机构	172
§ 9-4 基于 PC 机的多处理器系统	173
一、基于 PC 机的多处理器系统结构设想	173
二、基于 PC 机的多处理器系统的任务控制方法	174
三、多 CPU 板的硬件设计	175
四、基于 PC 机的多处理器系统软件的构成	176
习题与思考题	181
主要参考文献	183

第一章 概 述

§ 1-1 计算机技术的发展趋势

一、多处理机系统是科学发展的必然趋势

随着微电子技术的发展,特别是超大规模集成电路的发展,使得计算机的性能价格比有了很大的提高,从而进一步推动了计算机的应用和普及。但是,随着各行各业不断提出新的要求,如中长期的天气预报,对庞大数据库系统的检索,人工智能问题的求解等都要求新的计算机具有更高的速度。而传统的计算机结构已很难满足它们的要求,特别是在处理速度和系统可靠性方面尤为突出。从物理角度来看,限制计算机速度进一步提高的因素主要是门电路的延时。到1993年年底为止,当前速度最高的约瑟夫森逻辑门电路的延时时间约为7 ps。众所周知,光的传播速度是最高的,它传播1 mm、距离约要花3 ps时间。按此速度计算,门断开至接通的操作所花的时间,只使光传播2 mm的距离。另外,大规模集成电路的加工工艺及门电路的线间距离也随集成度的提高而出现了令人头痛的问题,例如,芯片的发热问题。一块芯片在工作时要消耗几十瓦的功率,这样就必须开发出耐温度差更大的新材料。这样一些问题会愈来愈突出。一句话,单个处理器的速度已慢慢地接近物理极限,要想再增加速度就要付出非常大的代价。

综观科学技术的发展史,我们不难发现这样一个规律,即一切技术的发展总是先从开始的集中的结构,而后再向分布结构过渡。例如,纺织工业在第一次工业革命以前,动力资源主要是水力。由于受地形、地物的限制,动力源水车采用集中的控制结构,用水车的主轴去驱动多台纺织机。但是,它受到水源提供的总能量及传输效率等方面的物理限制,使纺织机的速度和效率不能进一步提高。此后,随着蒸汽机、内燃机和发电机的出现,它们不仅使人类获得了高能量的能量源,而且使得纺织机有可能向分布结构方向发展。各台纺织机可配备独立的动力源,从而使能量损耗大幅度减小,机器的效率和可靠性也有了大幅度的提高。因此,与其它技术的发展趋势一样,计算机要提高其性能必须要避开物理上所出现的限制,采用新的技术(并行处理和分布结构),由集中控制结构向分布结构方向过渡,这是计算机技术发展的必由之路。

二、并行处理与多处理机系统

1. 单处理器系统的“并行处理”

并行处理的概念在单处理机系统中已有涉及,例如,多道程序的并行处理。但是这种“并行处理”的概念是以分时处理为基础的,实质上是多道程序的分时处理。每个程序只占一个时间片,用完分给的时间片后就暂停执行,然后再等待下一个分给的时间片的到来,再继续执行。4个程序按等分时间片“并行处理”的时序关系如图1-1所示。从图1-1中可以看出,尽管从宏观或从用户来看,4个程序在进行并行处理,但是从微观来看,处理器还是按顺序串行地一个程序一个程序地执行着。这种基于时间片的单处理器的“并行处理”不能算是真正的并行处理,它只能提高单处理器系统的效率,而不能大幅度提高系统的处理速度。

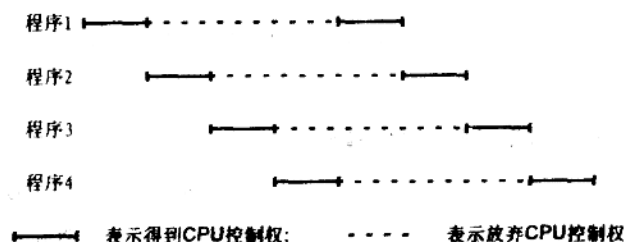


图 1-1 按时间片 4 个程序并行处理的时序关系

2. CPU 内部的并行处理

随着微电子技术的发展,人们为了提高微处理器或 CPU 的速度,在 CPU 内部配置了几个可以进行并行操作的部分。例如,在取当前指令的同时,对前一条已取出的指令进行解释,并开始执行再前一条已解释完毕的指令。3 条指令在 CPU 中是并行重叠处理的,这样就提高了 CPU 的处理速度。下面介绍一下在 CPU 内部进行并行处理的几种典型方式。

1) 向量计算机

向量处理机,从并行处理角度来看,它属于“运算流水线”类型。在向量运算中,尽管数据取值不同,但是运算的基本步骤是相同的。如果在向量运算的流水线中设置几个专用的运算单元,对数据进行流水作业处理,那么对数据可以实现并行重叠处理,从而使 CPU 的处理速度得到明显的提高。

这种向量处理机的典型例子是 CRAY-1。它是美国新墨西哥州的洛斯阿拉莫斯研究所于 1976 年 2 月研制出来的,采用时钟周期为 26.2 ns 的 ECL 器件,处理速度可达 160 MFLOPS(每秒执行百万次浮点运算)。图 1-2 是这种向量处理机的结构示意图。两个需要作相加运算的向量数据从左面进入运算管道。第一个运算单元对数据的阶数进行比较;第二个运算单元进行数的对阶;第三个运算单元作相加运算;第四个运算单元进行归一化处理。

这种 CPU 的结构特点是对于待处理的数据流采用专用硬件来进行流水作业处理,以提高 CPU 的速度。

2) 指令流水线处理机

指令流水线处理机与向量计算机非常类似。在这种 CPU 中,将执行指令的过程分成

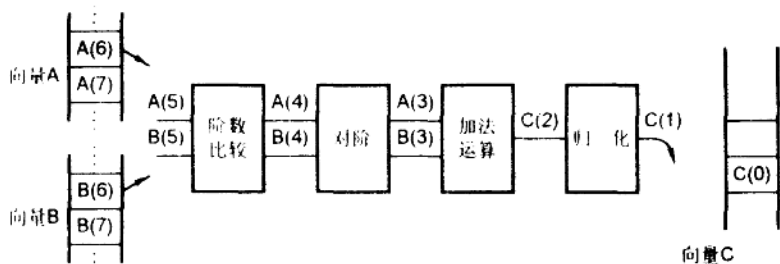


图 1-2 向量处理机的结构示意图

处理时间大致相等的几个步骤，如取指、执行和执行结果的存贮。这几个步骤的处理分别由专用的硬件来承担，以达到并行处理和提高 CPU 速度的目的。指令流水线处理机的示意图如图 1-3 所示。图 1-3 和图 1-2 的不同之处是，前者在管道中流动的是指令，而后者则是数据。

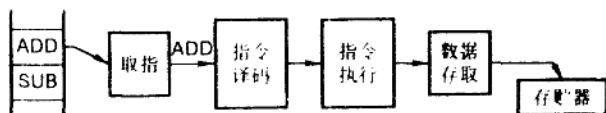


图 1-3 指令流水线处理机示意图

3) 多运算器的处理机

在这种处理机中配置了几个可以进行并行工作的浮点运算器，它们可以同时并行地执行互无关系的几条指令，其示意图如图 1-4 所示。

从图中可以看到，程序

$$A = B * C, D = \frac{E}{F}, G = H + I$$

3 部分可以分别由 3 个浮点运算器并行执行，从而提高了 CPU 的整体运算速度。

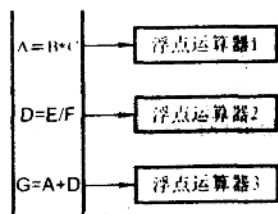


图 1-4 多运算器处理机示意图

4) 超级处理机

采用这种结构的处理机，如 Intel 公司的 80960CA 和 IBM 公司的 RS/6000，其结构与多运算器处理机的结构一样，多个运算器同时工作。所不同的是，处理机取出多个指令后，由硬件将这些指令分配给可并行工作的整数运算单元、浮点运算单元、代码存贮单元进行并行处理。

以上所举的 4 种处理机的共同特点是：对 CPU 的内部结构进行改造，使其具有并行处理能力，从而提高 CPU 的工作速度。但是这几种 CPU 所构成的计算机仍旧是一种单处理机（单 CPU）系统。我们今后研究的重点不是如何开发 CPU 内部的并行性，而是研究多个 CPU、多个处理机如何连接和进行并行处理，以此来提高系统的处理速度。

3. 多处理机系统

所谓多处理机系统,就是用户可以自由地使用系统中的多个 CPU 来并行地完成某个工作,工作中可以一边互相交换信息,一边协调地执行相应的程序。多处理机系统的两个典型例子如图 1-5 和图 1-6 所示。

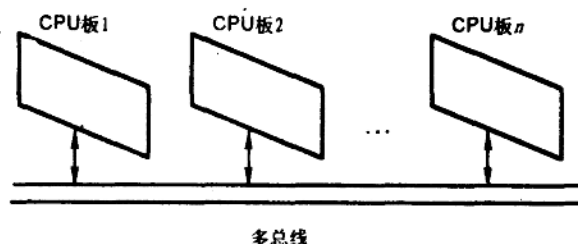


图 1-5 共享总线的多 CPU 系统



图 1-6 用网络连接的多微处理机系统

在图 1-5 中,多块 CPU 板通过多总线相连接,系统的编译程序将用户的源程序编译成可以并发执行的进程并分配给各个 CPU 执行。

图 1-6 是用网络连接的,由多个微型计算机构成的多处理机系统。如图所示,该系统是一个 CAD 系统,CAD 的程序模块可以并行地在各微型计算机中执行,从而提高了 CAD 系统的速度。

从上述两个典型例子可以看到,多处理机系统通常是用多个通用的 CPU 或通用的计算机,经某种拓扑连接而构成的系统。

§ 1-2 多处理机系统的一些基本概念

在最近的一些微型计算机广告中,常常可见到“该系统是一个多微处理器系统”这样一个词。在人们的心目中,一个微型计算机中如果包含两个以上的 CPU 就可以称之为多微处理器系统或多处理机系统。但是,这样的认识是笼统的,应该有一个比较确切的定义。

一、多处理机系统的定义和主要特征

所谓多处理机系统,就是由两个或两个以上的处理器或计算机,要么通过共享存储器,要么通过通信信道而连接起来的一个计算机系统。多处理机系统的结构示意图如图 1-7 所示。图中的方框可以是处理器(CPU)也可以是计算机,而连接网络可以是计算机的内部总线,也可以是其它各种类型的网络。

多处理机系统按系统连接来分, 可以分成紧耦合多处理机系统和松耦合多处理机系统。紧耦合多处理机系统通常是一个多 CPU, 共享存贮器连接的系统; 而松耦合多处理机系统较典型的是一个由多台计算机通过通信网络连接起来的系统, 一般又可以称为分布式计算机系统。由于系统结构不同, 因而这两种多处理机系统呈现不同的特征。

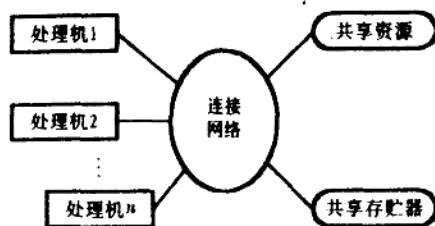


图 1-7 多处理机系统结构示意图

1. 紧耦合多处理机系统

根据美国学者 Enslow 提出的紧耦合的多处理机系统的主要特征是:

- (1) 一个系统至少包含两个以上的处理机(CPU);
- (2) 全部的处理机共享一个公共的存贮器;
- (3) 全部的处理机共用 I/O 通道, 控制单元和外设;
- (4) 在单一的操作系统支持下工作。

随着时间的推延和技术的发展, 上述这些特征发生了一些变化。如每个处理机可拥有本地存贮器和本地的 I/O 通道, 这些结构特征的改变主要是为了减少共享资源的冲突, 以进一步提高系统的效率。

2. 松耦合多处理机系统

松耦合多机系统的典型例子是由多台计算机组成的系统, 它一般可以称为分布式计算机系统。它的主要特征是:

(1) 系统中的任意两台计算机可以通过通信信道来交换信息。因此在系统中的任意两台计算机上的程序(或进程)可以使用系统提供的通信手段来交换数据。

(2) 系统中的各台计算机没有主次之分, 既没有控制整个系统的主机, 也没有受控于它机的从机。因此, 主、从控制计算机系统或分级控制计算机系统, 尽管也可以称之为松耦合的多机系统, 但是在技术上更倾向于单机系统, 或者说它们是由单机系统到多机系统的一种中间产品。不能将它们划入分布式计算机系统的范畴。

(3) 系统的资源为所有用户共享。在某台计算机上的用户不仅可以使位于该机的资源, 而且还可以使用位于它机上的资源。例如, 用户可以使用它机所配备的打印机来输出信息, 也可以访问存于它机的磁盘上的文件。分布式系统提供了资源共享的功能, 使得用户只需考虑系统是否具有所需的资源, 而无须考虑资源配置在哪一台计算机上。

(4) 系统中的若干台计算机可以互相协作来完成一个共同的任务, 或者说, 一个程序可以分布于几台计算机上并行地运行。

一般的计算机局域网应该说是一个由多个计算机连接起来的系统, 但是它决不是分布式计算机系统。图 1-8(a)、(b)都是用网络将多个计算机连接起来的系统。

图 1-8(a)是用以太网连接起来的计算机, 用户在各自的终端上作各自的工作, 程序在各自的计算机上运行, 尽管用户也可以共享磁盘上的文件和打印机, 但是它不能进行并行的程序处理, 故这种系统是一个计算机的局域网。

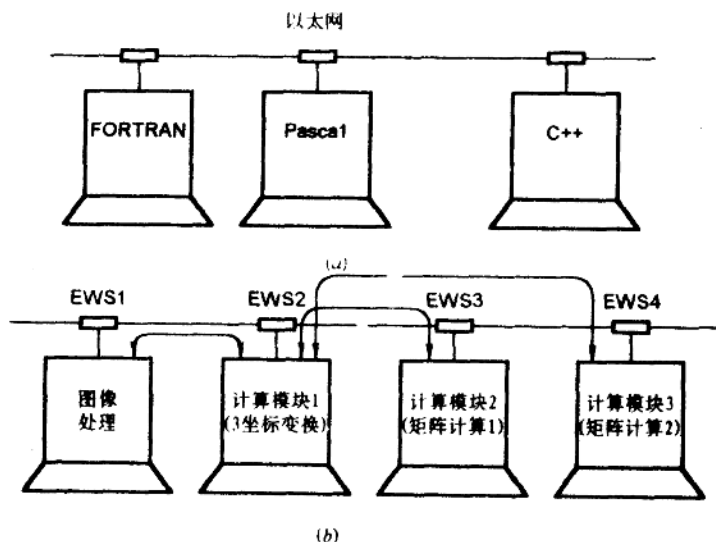


图 1-8 用网络连接计算机系统

(a) 计算机局域网; (b) 分布式计算机系统

图 1-8(b)也是用以太网连接起来的计算机系统。它是一套专用的 CAD 系统, 利用网络环境, 在 4 台计算机上可并行地执行 CAD 的相关程序, 这种系统可以称为多机系统或分布式计算机系统。

由此可知, 即使硬件的环境相同, 但是系统软件和使用方式不同, 其系统的功能也就不同。

从上面叙述我们可以看到, 无论是紧耦合多机系统还是松耦合多机系统, 它们都具备以下 3 个基本功能:

(1) 通信。系统应提供有效的通信手段, 使得运行于不同 CPU 或不同计算机上的用户程序可以通过通信来交换数据。

(2) 资源共享。系统应提供访问系统所有资源的功能。例如, 某个用户可以通过网络访问位于它机的资源。

(3) 并行运行用户程序。系统应提供并发程序设计语言和相应的编译程序, 使得用户编写的程序可以并行地在各 CPU 或计算机上执行。

二、并行处理计算机和多处理机系统的分类

并行处理计算机有多种分类方法。在这些分类中, 既包含 CPU 内部并行处理的计算机, 也包含多处理机系统。下面仅对几种常见分类方法作一介绍。

1. Flynn 分类方法

Flynn 分类方法是 M. J. Flynn 于 1966 年提出的一种并行处理的分类方法。它着眼于并行处理中的指令流和数据流的数目, 即处理的指令流和数据流是单一的(Single)还是多个的(Multiple)。例如, 一般的多处理机系统就属于 MIMD(多指令流多数据)。

1) SISD(Single Instruction Stream Single Data Stream: 单指令流单数据流)

该类计算机属于传统的冯·诺依曼计算机, 当前的大多数单机系统就属于此类。另

外,前面提到的指令流水线、多运算器及超级计算机也属于此类,其结构示意图如图 1-9 (a)所示。

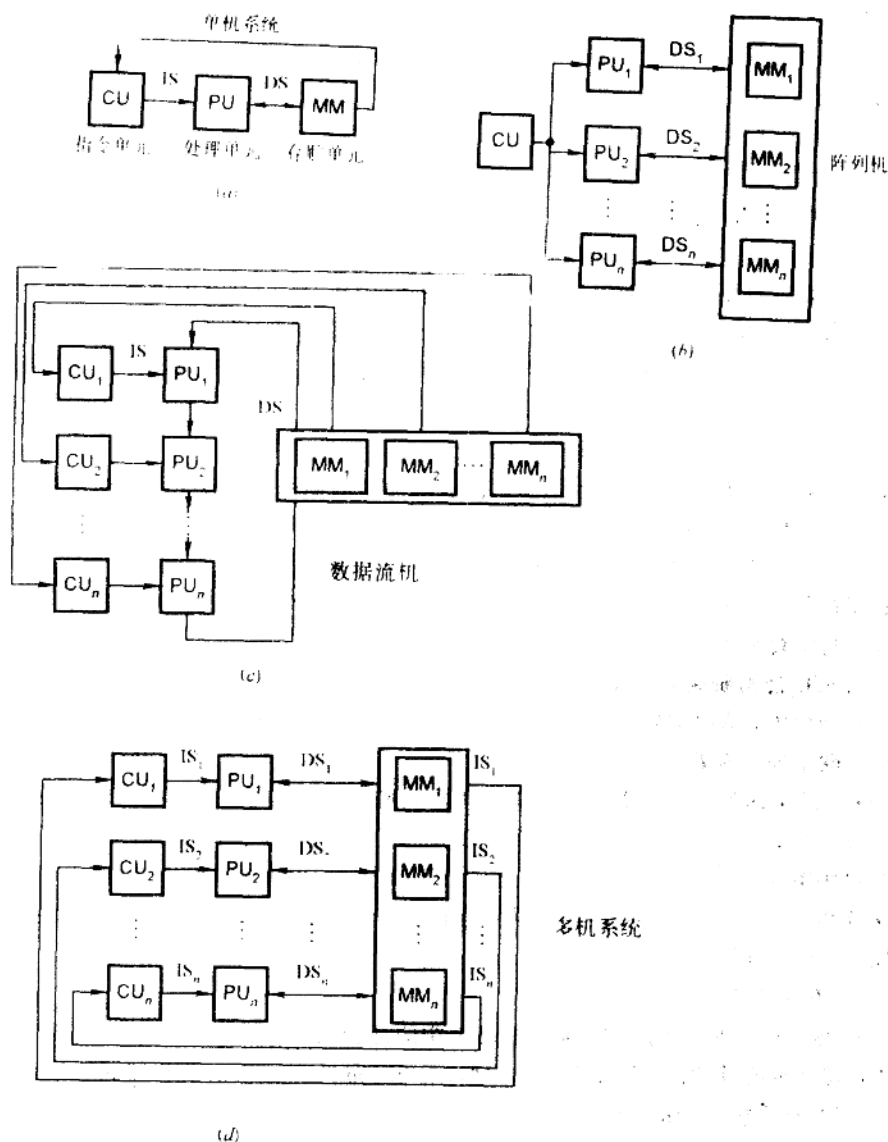


图 1-9 4 类计算机结构示意图

(a) SISD; (b) SIMD; (c) MISD; (d) MIMD

2) SIMD (Single Instruction Stream Multiple Data Stream; 单指令流多数据流)

阵列机属于此类计算机,指令单元发出的一条指令在多个处理单元得到执行,对多个数据进行处理。前面提到的向量计算机也属于此类。该类的结构示意图如图 1-9(b)所示。

3) MISD (Multiple Instruction Stream Single Data Stream; 多指令流单数据流)

数据流水线计算机属于此类计算机。数据顺序由多个处理单元,根据多个不同指令进行处理。该类的结构示意图如图 1-9(c)所示。

4) MIMD(Multiple Instruction Stream Multiple Data Stream: 多指令流多数据流)

典型的多处理机系统一般都属于该类,在此类系统中,多个 CPU(或多台计算机)各自并行地对多个数据进行处理。该类系统就是本书要重点介绍的内容,其结构示意图如图 1-9(d)所示。

2. Shore 分类方法

Shore 分类方法是 J. E. Shore 在 1973 年根据计算机控制结构提出来的一种分类方法,他将计算机的处理分成如下 6 类:

1) I 类计算机(字串行位并行)

该类计算机由单一的控制单元(CU)、处理单元(PU)、命令存贮器(M)和数据存贮器(DM)构成,是传统的冯·诺依曼计算机结构。当然 PU 可以由多个功能单元组成。这种计算机以字为单位顺序对数据进行处理。因此,指令流水线、流水线向量计算机等都属于此类计算机。其结构示意图如图 1-10(a)所示。

2) II 类计算机(字并行位串行)

这种计算机的 DM 由位片构成,PU 按位顺序对数据进行处理,即数据在存贮器中按一个字一行存贮,如果以位为单位来看,存贮器中的数据是一个二元矩阵。I 类计算机以水平位片为单位读入处理(即按字为单位);而 II 类计算机则以垂直位片为单位读入处理,例如数据存贮器为 1024×8 的二元矩阵,每个字为一个字节,那么在以垂直位片为单位,一次读入的位数为 1024 位。

美国 ICL 公司研制的 DAP(Distributed Array Processor)就属于这种类型。在信息检索这样一些领域中,有时只需要每个字中的几位信息。在这种情况下,一般的字串行位并行的计算机就不那么有效了。

另外,还考虑到联想存贮器和联想处理器的概念已经出现。这种结构有进一步推广应用的前景。例如,作为联想处理器的代表 Goodyear Aerospace 公司研制的 STRAN,它由 4 个阵列模块组成。各个阵列由 256 个 1 位的 PE(处理单元)构成。各 PE 之间由可塑性较大的网络连接,256 位的一个位片通过网络进行通信,并能同时进行通信。该类的结构示意图如图 1-10(b)所示。

3) III 类计算机(I 和 II 混合型)

在这类计算机中采用的结构是 I 类和 II 类的混合型,数据存贮器的二元阵列,无论按行,还是按列都能够读出,如图 1-10(c)所示。Shoeman 的正交计算机就属于该类计算机。

4) N 类计算机(非阵列连接)

该类计算机如图 1-10(d)所示。由 PU 和 DM 构成的处理单元 PE 只受独立的 CU 控制,特别是 PE 之间互相不进行通信。

5) V 类计算机(阵列连接)

该类计算机如图 1-10(e)所示。它支持邻接 PE 之间的通信。IAC 公司(Institute of Advanced Computation)的 ILLIAC N 就是该类计算机的典型代表。

6) VI 类计算机(处理器逻辑分散于存贮单元中)

该类计算机的处理器逻辑被分散在各存贮单元中,如图 1-10(f)所示。从简单的联想

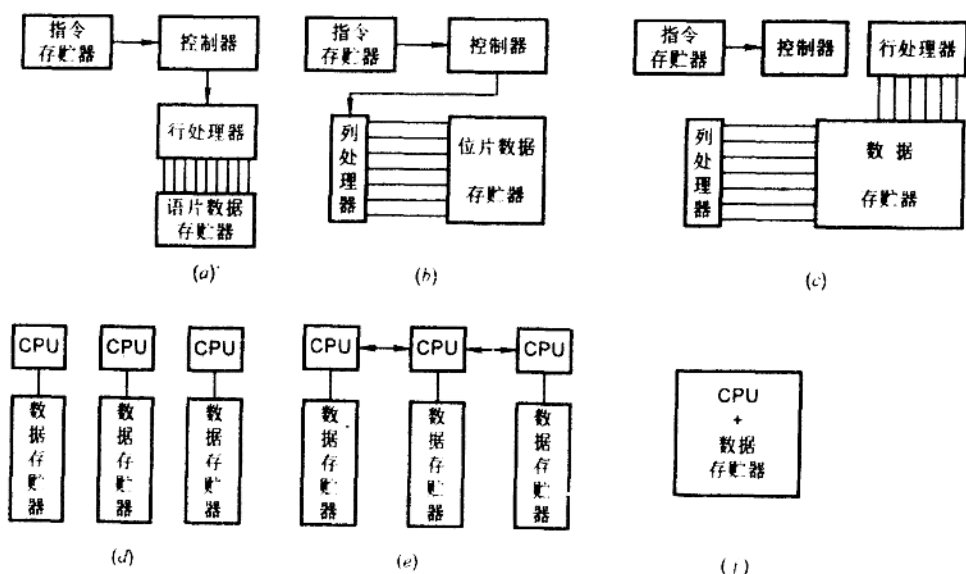


图 1-10 Shore 计算机分类

(a) I 类计算机：字串行位并行；(b) II 类计算机：字并行位串行；(c) I 类和 II 类的混合型；(d) 非阵列连接；(e) 阵列连接；(f) 处理器逻辑分散于存储单元中
存储器到复杂的联想处理器都属于这种结构。

由上述叙述可知，I 类计算机与 Flynn 的 SISD 类相对应，而 II 类到 V 类计算机则与 Flynn 的 SIMD 类相对应。

3. 按任务粒度的分类方法

从软件观点来看，并行处理是将一个作业分割成数个任务分配给各 CPU 执行。所分割任务的大小(任务粒度：Grain)可以有所不同。因此，也有根据任务粒度大小来对计算机进行分类的。

1) 粗粒度并行处理(Coarse Grain)

粗粒度并行处理，一般指对子程序或用户规定的较大的处理模块进行并行处理。众所周知，处理器之间的通信和同步的开销和任务大小有很大关系。因此，任务粒度大时，较适合后述的松耦合多机系统，这样系统处理的效率较高。另外，粒度愈大，系统中的任务数也愈少，从任务调度方面来看，实现起来也较容易，调度开销也较少，其示意图如图 1-11(a)所示。

2) 中粒度并行处理(Medial Grain)

这种处理与粗粒度并行处理相比任务稍小一点，例如 DO 循环语句的并行处理，循环体的大小就是并行处理任务的大小。图 1-11(b)就是其中一例。

在多 CPU 系统中，总是将一个比较短的公式作为一个任务粒度来考虑。在对连续多个公式进行处理时，前面的运算结果往往要在后面公式中用到。为了在分割任务时能保证数据流的连续性，一般要准备存放数据的 Cache(高速缓存)或本地存储器，这样不仅可以提高处理效率，而且还可以降低各 CPU 使用公共总线时的冲突概率。

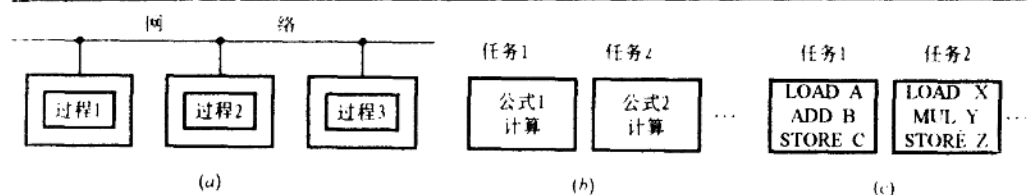


图 1-11 按粒度分类示意图

(a) 粗粒度并行处理：分割单位为过程级，任务粒度较大；

(b) 中粒度并行处理：分割单位按公式级别划分，任务粒度较小；

(c) 细粒度并行处理：分割单位按运算单位分割，任务粒度非常小

3) 细粒度并行处理(Fine Grain)

该类处理一般以指令为单位或接近于以指令为单位，这样细粒度的并行处理，如图 1-11(c)所示。从理论上讲，粒度小，CPU 空余的间隙也小，这有利于系统效率的提高。实现细粒度并行处理的实例如多运算单元计算机、数据流机和处理器阵列机等，它们都属于专用计算机。

在一般的通用多 CPU 系统中，如果粒度太小，并行处理的特有开销(同步处理和处理器之间的通信)就会增大，以至于大到不能忽视的程度。也就是说，任务数愈多，这些开销就愈大。对于共享总线或多机系统来说，效率就会大幅度降低。另外，任务数增加使得各任务如何分配给 CPU 也带来了较大困难，也就是说，任务调度的难度明显增加。

4. 按 CPU(或计算机)的耦合程度来分类的方法

这种分类方法是多处理机系统中常用的分类方法。在多处理机系统中，各处理机之间的通信路径带宽应该说是宽一点好，但是，这会随之带来系统价格的猛增。不仅如此，通信路径带宽还和前述的并行处理的粒度及处理机之间的耦合程度有关。也就是说，在设计多处理机系统时，应明确该系统的使用目标，并且必须确定采用几个 CPU(或几台计算机)，以及它们之间采用什么样的耦合方式。

在多处理机系统中，所谓松耦合和紧耦合，一般是指处理机之间的信息是用串行数据进行传送还是用并行数据进行传送。如果用串行数据交换信息，那么称这种多机系统为松耦合多机系统；若用并行数据进行信息交换，那么称这种多机系统为紧耦合多机系统。

1) 松耦合多 CPU(或多计算机)方式

松耦合多 CPU 系统的典型实例是由大量 CPU 用通信网络连接起来的分散存储器型多 CPU 系统，其结构如图 1-12(a)所示。图中 CPU 之间如采用串行通信，且通信速率为 10

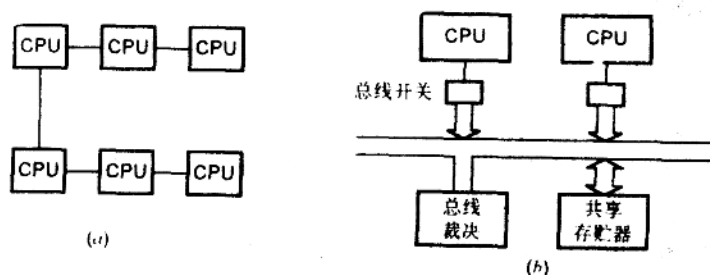


图 1-12 按 CPU(或计算机)耦合程度分类

(a) 松耦合多 CPU 系统； (b) 紧耦合多 CPU 系统

~20 M bit/s, 那么 CPU 之间只要一条信号线将它们连接起来就行了。一般为了降低系统价格, 总希望将 CPU 间的耦合程度尽可能降至最低。松耦合多机系统的结构一般比较简单, 但是根据网络连接的形态不同还可以分为静态网络和动态网络两种。所谓静态网络是指系统一旦完成, 其连接方式就不能改变。而动态网络一般是指系统在运行过程中由于某种需要可以改变系统的连接方式。

• 静态网络

典型静态网络的拓扑结构如图 1-13 所示。一维的是线性阵列; 二维的有矩阵、树形、环形和星形网络; 三维以上的有立方体、超立方体及完全连接网络等。

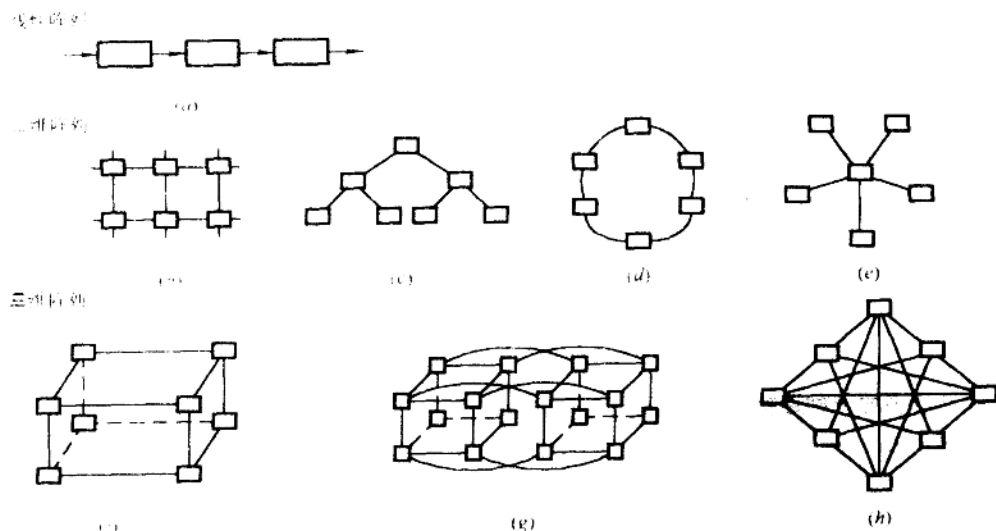


图 1-13 静态网络连接的分类

(a) 线性阵列; (b) 二维矩阵; (c) 树形; (d) 环形; (e) 星形;
(f) 立方体; (g) 超立方体; (h) 完全连接

• 动态网络

动态网络按串接的级数来分, 又可以分成单级网络和多级网络。所谓单级网络, 就是输入和输出的切换开关只有一级。例如图 1-14 就是用开关矩阵连接起来的多 CPU 系统。在单级连接网络中由源 CPU 到目的 CPU 只需经过一个开关, 这种结构比较适用于处理器之间数据传送比较频繁的情况。

多级网络在 CPU 连接的通道中必须通过几级开关, 由于有可能产生通信的竞争, 该网络路径选择度较高, 而且网络连接相对来说也较便宜。图 1-15 所示就是一个基准网络, 通过开关框的直通和交叉连接方式使得所有的 CPU 都能连接在一起。

以太网连接也是一种松耦合的网络连接方式, 如图 1-16 所示。在网上数据以包为单位进行信息交换, 所有的计算机通过一条共享的通信线传送信息。因此, 通信硬件价格便宜, 它适合于通信量较少的系统。如果通信量大, 而且非常频繁, 则采用开关网络较好。

2) 紧耦合多 CPU 方式

紧耦合多 CPU 系统的典型代表是多 CPU 共享总线连接的系统, 如图 1-17 所示。由于各 CPU 能像一般计算机一样对存贮器进行操作, 因此, 从编程角度来看比较简单, 是目