

目录

第 1 章 计算机系统结构的基本概念 1

第 2 章 计算机指令集结构设计 7

第 3 章 流水线技术 11

第 5 章 存储层次 15

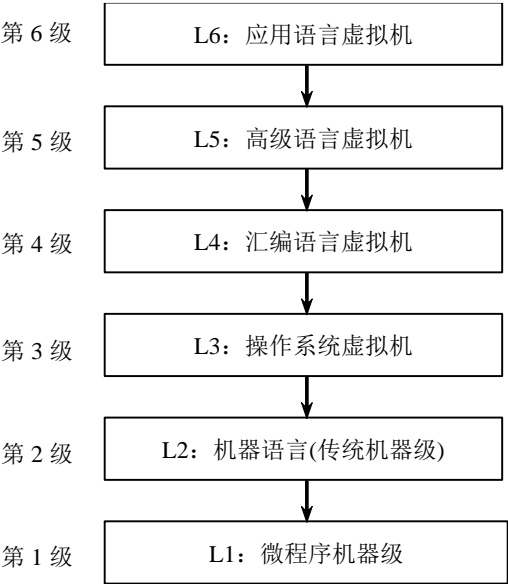
第 6 章 输入/输出系统 19

第 7 章 互连网络 22

第 8 章 多处理机 24

第 1 章 计算机系统结构的基本概念

1. 什么是计算机系统的多级层次结构？
从计算机语言的角度，把计算机系统按功能划分成以下多级层次结构：



2. 硬件和软件在什么意义上是等效的？在什么意义上是不等效的？
硬件和软件在功能实现上是等效的，即一种功能可以由软件实现，也可以由硬件实现。在实现性能上是不等效的。软件实现的优点是设计容易、改进简单；硬件实现的优点是速度

快。

3. 经典计算机系统结构的实质是什么？

计算机系统中软、硬件界面的确定，其界面之上的是软件的功能，界面之下的是硬件和固件的功能。

4. 语言实现的两种基本技术是什么？

翻译和解释是语言实现的两种基本技术。它们都是以执行一串 N 级指令来实现 $N+1$ 级指令，但二者存在着差别：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。

5. 对于通用寄存器型机器来说，机器语言程序设计者所看到的计算机的属性主要有哪些？

- (1) 数据表示
- (2) 寻址规则
- (3) 寄存器定义
- (4) 指令集
- (5) 中断系统
- (6) 机器工作状态的定义和切换
- (7) 存储系统
- (8) 信息保护
- (9) I/O 结构

6. 什么是软件兼容？软件兼容有几种？其中哪一种是软件兼容的根本特征？

同一个软件可以不加修改地运行于系统结构相同的各档机器上，而且它们所获得的结果一样，差别只在于运行时间的不同。

软件兼容分为向上兼容、向下兼容、向前兼容和向后兼容。其中向后兼容是软件兼容的根本特征。

7. 什么是系列机？它的出现较好地解决了什么矛盾？

系列机是指在一个厂家内生产的具有相同的系统结构，但具有不同组成和实现的一系列不同型号的机器。它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。

8. 对计算机发展非常关键的实现技术有哪些？

- (1) 逻辑电路
- (2) 半导体 DRAM(动态随机访问存储器)
- (3) 磁盘
- (4) 网络

9. 实现软件移植的主要途径有哪些？

- (1) 采用系列机方法

- (2) 采用模拟与仿真方法
- (3) 采用统一的高级语言

10. 试以系列机为例, 说明计算机系统结构、计算机组成和计算机实现三者之间的关系。

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。一种系统结构可以有多种组成; 一种组成可以有多种实现。同一系列机中各种型号的机器具有相同的系统结构, 但采用不同的组成和实现技术, 因而具有不同的性能和价格。

11. 存储程序计算机在系统结构上的主要特点是什么?

- (1) 以运算器为中心。
- (2) 采用存储程序原理。程序和数据放在同一存储器中, 并且没有对两者加以区分。
- (3) 存储器是按地址访问的、线性编址的空间。
- (4) 控制流由指令流产生。
- (5) 指令由操作码和地址码组成。操作码指明本指令的操作类型, 地址码指明操作数和操作结果的地址。
- (6) 数据以二进制编码表示, 采用二进制运算。

12. 从系统结构的发展情况看, 新型系统结构的设计主要从哪两方面着手?

新型系统结构的设计主要从以下两方面着手:

一方面是合理地增加计算机系统中硬件的功能比例, 使系统结构对操作系统、高级语言甚至应用软件提供更多更好的支持;

另一方面则是通过多种途径提高计算机系统结构中的并行性等级, 使得凡是能并行计算和处理的问题都能并行计算和处理, 使这种系统结构和组成对算法提供更多、更好的支持。

13. 软件技术两个最重要的发展趋势是什么?

- (1) 程序及数据所使用存储器容量的不断增大。
- (2) 高级语言的使用越来越广泛, 在很多应用领域取代了汇编语言。

14. 计算机系统设计人员的技术挑战主要来自哪几个方面?

计算机系统设计人员的技术挑战主要来自系统结构、设计工具、制造工艺、软件、应用和经济等多个方面。

15. 一种计算机系统结构的生命周期是怎样的?

计算机的生命周期和系统中各个部分的发展密切相关。一种新的系统结构的诞生, 往往以硬件为标志, 它的发展和成熟, 是以配套的系统软件和应用为标志的。一个计算机系统结构, 从产生到消亡, 大致需要 15~20 年的时间。

16. 用户 CPU 时间由哪三个因素决定?

用户 CPU 时间 = $CPI \times IC / \text{时钟频率}$

其中, CPI: 指令时钟数

IC: 程序执行过程中所处理的指令数。

17. 什么是测试程序组件? 在评价计算机系统设计时最常见的测试程序组件是哪个?

选择一组各个方面有代表性的测试程序，组成一个通用测试程序集合。这种测试程序集合称为测试程序组件。

在评价计算机系统设计时最常见的测试程序组件是基于 UNIX 的 SPEC，其主要版本包括 SPEC89、SPEC92、SPEC95 和 SPEC2000 等。

18. SPEC2000 测试程序组件中包括哪几个测试程序组件？

- (1) SPEC CPU2000: 测试 CPU;
- (2) SPECviewperf: 用于测试图形系统支持 OpenGL 库的性能;
- (3) SPECapc: 测试图形密集型应用的性能;
- (4) SPECsfs: 基于 NFS 文件系统的文件服务器测试程序;
- (5) SPECweb: Web 服务器测试程序。

19. 测试基于 Microsoft 公司的 Windows 系列操作系统平台的最常用测试组件有哪些？

- (1) PCMark04 包括中央处理器测试组、内存测试组、图形芯片测试组、硬盘测试组等。
- (2) Business Winstone 2004 主要用于测试计算机系统商业应用的综合性能。
- (3) Multimedia Content Creation Winstone 2004 主要用于测试计算机系统多媒体应用的综合性能。
- (4) SiSoft Sandra Pro 2004: 一套功能强大的系统分析评比工具，拥有超过 30 种以上的分析与测试模组，主要包括 CPU、存储器、I/O 接口、I/O 设备、主板等。

20. 常用的专门的性能指标测试程序有哪些？

3DMark03 主要测试显卡性能和 DirectX 的性能。

Prime95 是用来估计分布式计算程序的通信情况，可使计算机高负荷运转，所以也用来测试计算机的稳定性。

SuperPi/SuperE 是计算圆周率 π / 自然指数 e 的软件，通常用来测试 CPU 的稳定性。

21. 计算机系统结构设计和分析中最经常使用的三条基本原则是什么？

- (1) 大概率事件优先原则：对于大概率事件(最常见的事件)，赋予它优先的处理权和资源使用权，以获得全局的最优结果。
- (2) Amdahl 定律：加快某部件执行速度所获得的系统性能加速比，受限于该部件在系统所占的重要性。
- (3) 程序的局部性原理：程序在执行时所访问地址的分布不是随机的，而是相对地簇聚。

22. 根据 Amdahl 定律，系统加速比由哪两个因素决定？

系统加速比依赖于两个因素：

- (1) 可改进比例：可改进部分在原系统计算时间中所占的比例。
- (2) 部件加速比：可改进部分改进以后的性能提高。

23. 从执行程序的角度看，并行性等级从低到高可分为哪几级？

从执行程序的角度看，并行性等级从低到高可分为：

- (1) 指令内部并行：单条指令中各微操作之间的并行。
- (2) 指令级并行 (Instruction Level Parallelism, ILP)：并行执行两条或两条以上的指令。
- (3) 线程级并行 (Thread Level Parallelism, TLP)：并行执行两个或两个以上的线程，

通常是以一个进程内派生的多个线程为调度单位。

(4) 任务级或过程级并行：并行执行两个或两个以上的过程或任务（程序段），以子程序或进程为调度单元。

(5) 作业或程序级并行：并行执行两个或两个以上的作业或程序。

24. 从处理数据的角度，并行性等级从低到高可以分为哪几级？

从处理数据的角度，并行性等级从低到高可以分为：

- (1) 字串位串：同时只对一个字的一位进行处理。
- (2) 字串位并：同时对一个字的全部位进行处理。
- (3) 字并位串：同时对许多字的同一位（称位片）进行处理。
- (4) 全并行：同时对许多字的全部或部分位进行处理。

25. Flynn 分类法把计算机系统的结构分为哪 4 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）。

26. 常见的计算机系统结构分类法有哪两种？分类的依据是什么？

Flynn 分类法：按照指令流和数据流的多倍性进行分类。

冯氏分类法：用系统的最大并行度对计算机进行分类。

27. 计算机系统中提高并行性的技术途径有哪三种？

(1) 时间重叠。多个处理过程在时间上相互错开，轮流重叠地使用同一套硬件设备的各个部分，以加快硬件周转而赢得速度。

(2) 资源重复。通过重复设置资源，尤其是硬件资源，大幅度提高计算机系统的性能。

(3) 资源共享。这是一种软件方法，它使多个任务按一定时间顺序轮流使用同一套硬件设备。

28. 多机系统的耦合度可以分为哪几类？

(1) 最低耦合：除通过某种中间存储介质之外，各计算机之间没有物理连接，也无共享的联机硬件资源。

(2) 松散耦合：通过通道或通信线路实现计算机间互连，共享某些外围设备，机间的相互作用是在文件或数据集一级进行。

(3) 紧密耦合：机间物理连接的频带较高，往往通过总线或高速开关实现互连，可以共享主存。

29. 单机系统和多机系统中，都是按哪三种技术途径分别发展为哪三类多处理机？

单机系统和多机系统中，都是按时间重叠、资源重复和资源共享三种技术途径，分别发展为同构型多处理机、异构型多处理机、分布处理系统。

30. 三种类型的多处理机（同构型多处理机、异构型多处理机、分布处理系统）的主要区别是什么？

项目	同构型多处理机	异构型多处理机	分布处理系统
目的	提高系统性能 (可靠性、速度)	提高系统使用效率	兼顾效率与性能
技术 途径	资源重复 (机向互连)	时间重叠 (功能专用化)	资源共享 (网络化)
组成	同类型 (同等功能)	不同类型 (不同功能)	不限制
分工方式	任务分布	功能分布	硬件、软件、数据等各种资源分布
工作方式	一个作业由多机 协同并行地完成	一个作业由多机 协同串行地完成	一个作业由一台处理机完成，必要时才请求它机协作
控制形式	常采用浮动控制方式	采用专用控制方式	分布控制方式
耦合度	紧密耦合	紧密、松散耦合	松散、紧密耦合
对互连网络的要求	快速性、灵活性、可重构性	专用性	快速、灵活、简单、通用

第 2 章 计算机指令集结构设计

1. 通常可按哪 5 个因素对计算机指令集结构进行分类？
- (1) 在 CPU 中操作数的存储方法。
 - (2) 指令中显式表示的操作数个数。
 - (3) 操作数的寻址方式。
 - (4) 指令集所提供的操作类型。
 - (5) 操作数的类型和大小。
2. 在对计算机指令集结构进行分类的 5 个因素中，哪一种是各种指令集结构之间最主要的区别？
- CPU 中操作数的存储方法，即在 CPU 中用来存储操作数的存储单元的类型，是各种指令集结构之间最主要的区别。
3. 根据 CPU 内部存储单元类型，可将指令集结构分为哪几类？
- 堆栈型指令集结构、累加器型指令集结构、通用寄存器型指令集结构。
4. 堆栈型指令集结构、累加器型指令集结构和通用寄存器型指令集结构分别有什么优缺点？

指令集结构类型	优点	缺点
堆栈型	是一种表示计算的简单模型；指令短小	不能随机访问堆栈，从而很难生成有效代码。同时，由于堆栈是瓶颈，所以很难被高效地实现
累加器型	减少了机器的内部状态；指令短小	由于累加器是唯一的暂存器，这种机器的存储器通信开销最大
寄存器型	易于生成高效的目标代码	所有操作数均需命名，且要显式表示，因而指令比较长

5. 现代大多数机器均采用通用寄存器型指令集结构，为什么？
- 主要有两个方面的原因，一是寄存器和 CPU 内部其他存储单元一样，要比存储器快；其次是对编译器而言，可以更加容易、有效地分配和使用寄存器。
6. 通用寄存器型指令集结构可细分为哪三类？
- 寄存器－寄存器型。
 - 寄存器－存储器型。
 - 存储器－存储器型。
7. 三种通用寄存器型指令集结构分别有什么优缺点？

指令集结构类型	优 点	缺 点
寄存器－寄存器型 (0,3)	简单，指令字长固定，是一种简单的代码生成模型，各种指令的执行时钟周期数相近	和 ALU 指令中含存储器操作数的指令集结构相比，指令条数多，因而其目标代码量较大
寄存器－存储器	可以直接对存储器操作数进行	指令中的操作数类型不同。在一条指令中同时对

(1,2)	访问，容易对指令进行编码，且其目标代码量较小	一个寄存器操作数和存储器操作数进行编码，将限制指令所能够表示的寄存器个数。由于指令的操作数可以存储在不同类型的存储器单元，所以每条指令的执行时钟周期数也不尽相同
存储器 - 寄存器型 (3,3)	是一种最紧密的编码方式，无需“浪费”寄存器保存变量	指令字长多种多样。每条指令的执行时钟周期数也大不一样，对存储器的频繁访问将导致存储器访问瓶颈问题

8. 从当前的计算机技术观点来看，CISC 结构有什么缺点？

- (1) CISC 结构的指令系统中，各种指令的使用频率相差悬殊。
- (2) CISC 结构指令系统的复杂性带来了计算机系统结构的复杂性，这不仅增加了研制时间和成本，而且还容易造成设计错误。
- (3) CISC 结构指令系统的复杂性给 VLSI 设计增加了很大负担，不利于单片集成。
- (4) CISC 结构的指令系统中，许多复杂指令需要很复杂的操作，因而运行速度慢。
- (5) 在 CISC 结构的指令系统中，由于各条指令的功能不均衡性，不利于采用先进的计算机系统结构技术来提高系统的性能。

9. 增强 CISC 机器的指令功能主要从哪几方面着手？

- (1) 面向目标程序增强指令功能。
- (2) 面向高级语言和编译程序改进指令系统。
- (3) 面向操作系统的优化实现改进指令系统。

10. RISC 的设计原则是什么？

- (1) 选取使用频率最高的指令，并补充一些最有用的指令。
- (2) 每条指令的功能应尽可能简单，并在一个机器周期内完成。
- (3) 所有指令长度均相同。
- (4) 只有 load 和 store 操作指令才访问存储器，其它指令操作均在寄存器之间进行。
- (5) 以简单、有效的方式支持高级语言。

11. RISC 和 CISC 处理机的指令系统结构在指令格式、寻址方式和每条指令的周期数 (CPI) 三方面有哪些区别？

比较内容	CISC	RISC
指令格式	变长编码	定长编码
寻址方式	各种都有	只有 load/store 指令可以访存
CPI	远远大于 1	为 1

12. 计算机指令集结构设计所涉及的内容有哪些？

- (1) 指令集功能设计：主要有 RISC 和 CISC 两种技术发展方向。
- (2) 寻址方式的设计。
- (3) 操作数表示和操作数类型。
- (4) 寻址方式的表示：可以将寻址方式编码于操作码中，也可以将寻址方式作为一个单独的域来表示。

(5) 指令集格式的设计：有变长编码格式、固定长度编码格式和混合型编码格式三种。

13. 指令中有哪两种表示操作数类型的方法？

- (1) 操作数的类型由操作码的编码指定，这是最常见的一种方法。
- (2) 数据可以附上由硬件解释的标记，由这些标记指定操作数的类型，从而选择适当的运算。

14. 指令中表示寻址方式的主要方法有哪些？

表示寻址方式有两种常用的方法：

(1) 将寻址方式编于操作码中，操作码在描述指令功能的同时也描述了相应的寻址方式。这种方式译码快，但操作码和寻址方式的结合不仅增加了指令的条数，导致了指令的多样性，而且增加了 CPU 对指令译码的难度。

(2) 为每个操作数设置一个地址描述符，由该地址描述符表示相应操作数的寻址方式。这种方式译码较慢，但操作码和寻址独立，易于指令扩展。

15. 指令集格式设计中选择表示寻址方式的依据是什么？

主要由指令集结构所采用的寻址方式种类及其适用范围，以及操作码与寻址方式之间的独立程度来决定。

如果某些指令集结构的指令有 1~5 个操作数，每个操作数有 10 种寻址方式，对于这种大规模的操作数和寻址方式组合，通常采用增设地址描述符的方法来描述寻址方式。

对诸如 load/store 类型指令集结构的指令，由于只有 1~3 个操作数，而且只有有限几种寻址方式。通常将寻址方式编码于操作码中。

16. 指令集结构中采用多种寻址方式有何优缺点？

在指令集结构中采用多种寻址方式可以显著地减少程序的指令条数；但同时也可能增加实现的复杂度和使用这些寻址方式的指令的执行时钟周期数（CPI）。

17. 在控制指令中使用 PC 相对寻址方式有什么优点？

- (1) 有效地缩短指令中表示目标地址的字段的长度。
- (2) 使得代码在执行时与它被载入的位置无关。

18. 在指令集格式的设计时，通常可选择哪几种指令格式？

- (1) 变长编码格式。这种编码方式可以有效减少程序的目标代码大小。
- (2) 定长编码格式。这种编码方式可以降低译码的复杂度，提高译码的性能。
- (3) 混合型编码格式。兼顾降低目标代码长度和降低译码复杂度。

19. MIPS 中有哪些寄存器？

- (1) 32 个通用寄存器。
- (2) 32 个浮点寄存器。
- (3) 一些特别的寄存器。

20. MIPS 有哪几种数据类型？

- (1) 整数：字节（8 位），半字（16 位），字（32 位）和双字（64 位）。

(2) 浮点数：单精度浮点数（32 位），双精度浮点数（64 位）。

21. MIPS 采用哪几种寻址方式？

MIPS 的数据寻址方式只有立即数寻址和偏移量寻址两种，立即数字段和偏移量字段都是 16 位的。寄存器间接寻址是通过把 0 作为偏移量来实现的，16 位绝对寻址是通过把 R0（其值永远为 0）作为基址寄存器来完成的。

第3章 流水线技术

1. 流水技术有哪些特点？

- (1) 流水过程由多个相联系的子过程组成，每个过程称为流水线的“级”或“段”。
- (2) 每个子过程由专用的功能段实现。
- (3) 各个功能段所需时间应尽量相等。
- (4) 流水线需要有“通过时间”，在此之后流水过程才进入稳定工作状态，每一个时钟周期(拍)流出一个结果。
- (5) 流水技术适合于大量重复的时序过程，只有在输入端能连续地提供任务，流水线的效率才能充分发挥。

2. 按照流水线所完成的功能来分，流水线可分为哪两类？

- (1) 单功能流水线：只能完成一种固定功能的流水线。
- (2) 多功能流水线：流水线的各段可以进行不同的连接，从而使流水线在不同的时间，或者在同一时间完成不同的功能。

3. 按照同一时间内各段之间的连接方式来分，流水线可分为哪两类？

- (1) 静态流水线：在同一时间内，流水线的各段只能按同一种功能的连接方式工作。
- (2) 动态流水线：在同一时间内，当某些段正在实现某种运算时，另一些段却在实现另一种运算。

4. 按照流水的级别来分，流水线可分为哪三类？

- (1) 部件级流水线（运算操作流水线）：把处理机的算术逻辑部件分段，以便为各种数据类型进行流水操作。
- (2) 处理机级流水线（指令流水线）：把解释指令的过程按照流水方式处理。
- (3) 处理机间流水线（宏流水线）：由两个以上的处理机串行地对同一数据流进行处理，每个处理机完成一项任务。

5. 按照数据表示来分，流水线可分为哪两类？

- (1) 标量流水处理机：处理机不具有向量数据表示，仅对标量数据进行流水处理。
- (2) 向量流水处理机：处理机具有向量数据表示，并通过向量指令对向量的各元素进行处理。

6. 按照流水线中是否有反馈回路来分，流水线可分为哪两类？

- (1) 线性流水线：流水线的各段串行连接，没有反馈回路。
- (2) 非线性流水线：流水线中除有串行连接的通路外，还有反馈回路。

7. 在5段流水线中，一条指令的执行需要几个时钟周期？它们分别是什么？

一条指令的执行需要5个时钟周期。它们分别是：取指令周期（IF）、指令译码/读寄存器周期（ID）、执行/有效地址计算周期（EX）、存储器/分支完成周期（MEM）、写回周期（WB）。

8. 流水线寄存器的作用是什么？

把数据和控制信息从一个流水段传送到下一个流水段。

9. 消除流水线的瓶颈段有哪些方法？

- (1) 细分瓶颈段；
- (2) 重复设置瓶颈段。

10. 评价流水线的性能指标是什么？

- (1) 吞吐率：指在单位时间内流水线所完成的任务数或输出结果的数量。
- (2) 流水线的加速比：指 m 段流水线的速度与等功能的非流水线的速度之比。
- (3) 效率：指流水线的设备利用率。

11. 什么叫相关？流水线中有哪几种相关？

相关是指两条指令之间存在某种依赖关系。确定程序中指令之间存在什么样的相关，对于充分发挥流水线的效率有重要的意义。

相关有 3 种类型，分别是数据相关（也称真数据相关）、名相关、控制相关。

12. 什么叫名相关？名相关有哪两种？

如果两条指令使用相同的名，但是它们之间并没有数据流动，则称这两条指令存在名相关。

指令 j 与指令 i 之间的名相关有以下两种：

- (1) 反相关。如果指令 j 写的名与指令 i 读的名相同，则称指令 i 和 j 发生了反相关。反相关指令之间的执行顺序是必须严格遵守的，以保证 i 读的值是正确的。
- (2) 输出相关。如果指令 j 和指令 i 写相同的名，则称指令 i 和 j 发生了输出相关。输出相关指令的执行顺序是不能颠倒的，以保证最后的结果是指令 j 写进去的。

13. 流水线冲突有哪几种？

流水线冲突有以下 3 种类型：

- (1) 结构冲突：因硬件资源满足不了指令重叠执行的要求而发生的冲突。
- (2) 数据冲突：当指令在流水线中重叠执行时，因需要用到前面指令的执行结果而发生的冲突。
- (3) 控制冲突：流水线遇到分支指令和其他会改变 PC 值的指令所引起的冲突。

14. 解决流水线结构冲突的方法有哪些？

- (1) 流水化功能单元；
- (2) 资源重复；
- (3) 暂停流水线。

15. 为什么流水线设计者有时会允许结构冲突的存在？

主要有两个原因：一是为了减少硬件开销，二是为了减少功能单元的延迟。

16. 定向技术的主要思想是什么？

在发生数据冲突时，后面的指令并不是立即就要用到前一条指令的计算结果。如果能够将其计算结果从其产生的地方直接送到需要它的地方，就可以避免暂停。当定向硬件检测到前面某条指令的结果寄存器就是当前指令的源寄存器时，控制逻辑会将前面那条指令的结果直接从其产生的地方定向到当前指令所需的位置。

17. 流水线中所有数据冲突都可以通过定向技术消除，而不需要暂停吗？试举例说明。

流水线中有需要暂停的数据冲突。

例如，在下列指令序列中，需要在 LW 指令之后插入一个暂停周期，然后再用定向技术消除相关。

```
LD      R1, 0 (R2)
DADD    R4, R1, R5
AND      R6, R1, R7
XOR      R8, R1, R9
```

18. 根据指令对寄存器的读写顺序，可将数据冲突分为哪三种类型？

- (1) 写后读冲突
- (2) 写后写冲突
- (3) 读后写冲突

19. 解决流水线数据冲突的方法有哪些？

(1) 定向技术：在某条指令产生一个结果之前，其他指令并不真正需要该计算结果，如果将该计结果从其产生的地方直接送到其他指令需要它的地方，就可以避免暂停。

(2) 暂停技术：设置一个“流水线互锁”的功能部件，一旦流水线互锁检测到数据相关，流水线暂停执行发生数据相关指令后续的所有指令，直到该数据相关解决为止。

- (3) 采用编译器调度。
- (4) 重新组织代码顺序。

20. 减少流水线处理分支指令时的暂停时钟周期数有哪两种途径？

- (1) 在流水线中尽早判断出分支转移是否成功。
- (2) 尽早计算出分支转移成功时的 PC 值（即分支的目标地址）。

21. 在 MIPS 基本流水线中可采用哪些静态方法降低分支损失？

- (1) 修改硬件：在 ID 段增加一个加法器，将计算分支目标地址的操作移到 ID 段完成。
- (2) 预测分支失败的方法。
- (3) 延迟分支方法。

22. 从编译技术的角度，降低流水线分支损失的方法有哪些？

- (1) 预测分支失败的方法。
- (2) 预测分支成功的方法。
- (3) 延迟分支方法。

23. 预测分支失败方法的主要思想是什么？

当流水线译码到一条分支指令时，流水线继续取指令，并允许该分支指令后的指令继续在流水线中流动。当流水线确定分支转移成功与否以及分支的目标地址之后，如果分支转移成功，流水线必须将在分支指令之后取出的所有指令转化为空操作，并在分支的目标地址处重新取出有效的指令；如果分支转移失败，那么可以将分支指令看作是一条普通指令，流水线正常流动，无需将在分支指令之后取出的所有指令转化为空操作。

23. 预测分支成功方法的主要思想是什么？

预测分支成功方法的主要思想是：

假设分支转移成功，并从分支目标地址处取指令执行。当流水线译码 ID 段检测到分支指令后，一旦计算出了分支目标地址，就开始从该目标地址取指令执行。起作用的前提：先知道分支目标地址，后知道分支是否成功。前述 5 段流水线中，这种方法没有任何好处，因为判断分支是否成功与分支目标地址计算是在同一流水段完成的。

24. “延迟分支”方法的主要思想是什么？

其主要思想是从逻辑上“延长”分支指令的执行时间。设延迟长度为 n 的分支指令后面有 n 个分支延迟槽，选择 n 条有效和有用的指令放入分支延迟槽中，无论分支成功与否，流水线都会执行这些指令。处于分支延迟槽中的指令“掩盖”了流水线原来所必须插入的暂停周期。

25. 为了在分支延迟槽中填入有效指令，一般采用哪三种方法？

从前调度、从目标处调度、从失败处调度

26. 调度分支延迟指令的三种常用方法的特点和局限性是什么？

调度策略	对调度的要求	对流水线性能改善的影响
从前调度	分支必须不依赖于被调度的指令	总是可以有效提高流水线性能
从目标处调度	如果分支转移失败，必须保证被调度的指令对程序的执行没有影响，可能需要复制被调度指令	分支转移成功时，可以提高流水线性能。但由于复制指令，可能加大程序空间
从失败处调度	如果分支转移成功，必须保证被调度的指令对程序的执行没有影响	分支转移失败时，可以提高流水线性能

27. 有哪几种向量处理方式？它们对向量处理机的结构要求有何不同？

(1) 水平处理方式：不适合对向量进行流水处理。

(2) 垂直处理方式：适合对向量进行流水处理，向量运算指令的源/目向量都放在存储器内，使得流水线运算部件的输入、输出端直接与存储器相联，构成 MM 型的运算流水线。

(3) 分组处理方式：适合流水处理。可设长度为 n 的向量寄存器，使每组向量运算的源/目向量都在向量寄存器中，流水线的运算部件输入、输出端与向量寄存器相联，构成 RR 型运算流水线。

28. 什么是向量链接技术？

当两条向量指令出现“写后读”相关时，若它们不存在功能部件冲突和向量寄存器(源或目的)冲突，就有可能把它们所用的功能部件头尾相接，形成一个链接流水线，进行流水处理。

29. 衡量向量处理机性能的主要参数有哪些？

(1) 向量指令的处理时间。

(2) 峰值性能：向量长度为无穷大时，向量处理机的最高性能。

(3) 半性能向量长度：向量处理机的性能为其峰值性能一半时所需的向量长度。

(4) 向量长度临界值：对同一段程序代码而言，向量方式的处理速度优于标量串行方式处理速度时所需的最小向量长度。

第5章 存储层次

1. 单级存储器的主要矛盾是什么？通常采取什么方法来解决？

主要矛盾：

- (1) 速度越快，每位价格就越高。
- (2) 容量越大，每位价格就越低。
- (3) 容量越大，速度越慢。

采取多级存储层次方法来解决。

2. 评价存储层次的主要参数有哪些？

存储层次的平均每位价格、命中率或失效率、平均访问时间。

3. “Cache-主存”和“主存-辅存”层次的主要区别是什么？

比较项目 \ 存储层次	“Cache-主存”层次	“主存-辅存”层次
目的	为了弥补主存速度的不足	为了弥补主存容量的不足
存储管理的实现	全部由专用硬件实现	主要由软件实现
访问速度的比值 (第一级比第二级)	几比一	几万比一
典型的块(页)大小	几十个字节	几百到几千个字节
CPU 对第二级的访问方式	可直接访问	均通过第一级
不命中时 CPU 是否切换	不切换	切换到其他进程

4. 在存储层次中应解决哪四个问题？

- (1) 映像规则：当把一个块调入高一层存储器时，可以放到哪些位置上。
- (2) 查找算法：当所要访问的块在高一层存储器中时，如何找到该块。
- (3) 替换算法：当发生失效时，应替换哪一块。
- (4) 写策略：当进行写访问时，应进行哪些操作。

5. 地址映像方法有哪几种？它们各有什么优缺点？

(1) 全相联映像。实现查找的机制复杂，代价高，速度慢。Cache 空间的利用率较高，块冲突概率较低，因而 Cache 的失效率也低。

(2) 直接映像。实现查找的机制简单，速度快。Cache 空间的利用率较低，块冲突概率较高，因而 Cache 的失效率也高。

(3) 组相联映像。组相联是直接映像和全相联的一种折中。

6. 组相联 Cache 比相同容量的直接映像 Cache 的失效率低。由此是否可以得出结论：采用组相联 Cache 一定能带来性能上的提高？为什么？

不一定。因为组相联命中率的提高是以增加命中时间为代价的，组相联需要增加多路选择开关。

7. Cache 中，有哪两种实现并行查找的方法？

- (1) 用相联存储器实现。
- (2) 用单体多字存储器和比较器来实现。

8. 替换算法有哪几种？它们各有什么优缺点？

(1) 随机法：简单、易于用硬件实现，但这种方法没有考虑 Cache 块过去被使用的情况，反映不了程序的局部性，所以其失效率比 LRU 的高。

(2) 先进先出法：容易实现。它虽然利用了同一组中各块进入 Cache 的顺序这一“历史”信息，但还是不能正确地反映程序的局部性。

(3) 最近最少使用法 LRU：失效率最低。但是 LRU 比较复杂，硬件实现比较困难。

9. 写策略主要有哪两种？它们各有什么优点？

(1) 写直达法：易于实现，而且下一级存储器中的数据总是最新的。

(2) 写回法：速度快，写操作能以 Cache 存储器的速度进行。而且对于同一单元的多个写最后只需一次写回下一级存储器，有些“写”只到达 Cache，不到达主存，因而所使用的存储器频带较低。

10. 在写回法中，可采用什么方法减少在替换时块的写回？

常采用“污染位”标志。即为 Cache 中的每一块设置一个“污染位”（设在与该块相应的目录表项中），用于指出该块是“脏”的（被修改过）还是干净的（没被修改过）。替换时，若被替换的块是干净的，则不必写回下一级存储器，因为这时下一级存储器中相应块的内容与 Cache 中的一致。

11. 当发生 Cache 写失效时，是否调入相应的块，有哪两种选择？

(1) 按写分配法：写失效时，先把所写单元所在的块调入 Cache，然后再进行写入。这种方法也称为写时取方法。

(2) 不按写分配法：写失效时，直接写入下一级存储器而不将相应的块调入 Cache。这种方法也称为绕写法。

12. 按照 Cache 产生失效的原因不同，可以把失效分为哪三类？

强制性失效、容量失效、冲突失效。

13. 3C 失效与 Cache 的相联度、容量有什么样的关系？

(1) 相联度越高，冲突失效就越少。

(2) 强制性失效和容量失效不受相联度的影响。

(3) 强制性失效不受 Cache 容量的影响，但容量失效却随着容量的增加而减少。

(4) 2:1 的 Cache 经验规则：大小为 N 的直接映像 Cache 的失效率约等于大小为 $N/2$ 的两路组相联 Cache 的失效率。

14. 增加 Cache 块大小一定会降低失效率吗？

不一定。

对于给定的 Cache 容量，当块大小增加时，失效率开始是下降，后来反而上升了。主要因为增加块大小会产生双重作用。一方面它减少了强制性失效；另一方面，可能会增加冲突失效。

15. 伪相联的基本思想是什么？

采用这种方法时，在命中情况下，访问 Cache 的过程和直接映像 Cache 中的情况相同；而发生失效时，在访问下一级存储器之前，会先检查 Cache 另一个位置（块），看是否匹配。确定这个另一块的一种简单的方法是将索引字段的最高位取反，然后按照新索引去寻找伪相联组中的对应块。如果这一块的标识匹配，则称发生了伪命中。否则，就只好访问下一级存储器。

16. 伪相联的优点是什么？

伪相联既能获得多路组相联 Cache 的低失效率又能保持直接映像 Cache 的命中速度。

17. 降低 Cache 失效率有哪些方法？

增加 Cache 块大小、提高相联度、增加 Cache 的容量、Victim Cache、伪相联 Cache、硬件预取技术、由编译器控制的预取、编译器优化。

18. 减少 Cache 失效开销有哪些方法？

- (1) 让读失效优先于写。
- (2) 写缓冲合并。
- (3) 请求字处理技术。
- (4) 非阻塞 Cache 或非锁定 Cache 技术。
- (5) 采用二级 Cache。

19. 减少 Cache 命中时间有哪些方法？

容量小且结构简单的 Cache、虚拟 Cache、Trace Cache、Cache 访问流水化。

20. 请求字处理技术有哪两种具体的实现方法？

(1) 尽早重启动：在请求字没有到达时，CPU 处于等待状态。一旦请求字到达，就立即发送给 CPU，让等待的 CPU 尽早重启动，继续执行

(2) 请求字优先：调块时，首先向存储器请求 CPU 所要的请求字。请求字一旦到达，就立刻送往 CPU，让 CPU 继续执行，同时从存储器调入该块的其余部分。请求字优先也称为回绕读取或关键字优先。

21. 采用二级 Cache 的基本思想是什么？

通过在原有 Cache 和存储器之间增加另一级 Cache，构成两级 Cache。把第一级 Cache 做得足够小，使其速度和快速 CPU 的时钟周期相匹配，而把第二级 Cache 做得足够大，使它能捕获更多本来需要到主存去的访问，从而降低实际失效开销。

22. 采用容量小且结构简单的 Cache 有什么好处？

(1) 可以有效地提高 Cache 的访问速度。因为硬件越简单，速度就越快。小容量 Cache 可以实现快速标识检测，对减少命中时间有益。

(2) Cache 足够小, 可以与处理器做在同一芯片上, 以避免因芯片外访问而增加时间开销。

(3) 保持 Cache 结构简单可采用直接映像 Cache。直接映像 Cache 的主要优点是可以让标识检测和数据传送重叠进行, 这样可以有效地减少命中时间。

23. “虚拟索引 + 物理标识” Cache 的基本思想是什么?

直接用虚地址中的页内位移 (页内位移在虚实地址的转换中保持不变) 作为访问 Cache 的索引, 但标识却是物理地址。CPU 发出访存请求后, 在进行虚实地址转换的同时, 可并行进行标识的读取。在完成地址变换之后, 再把得到的物理地址与标识进行比较。

24. 可采用什么方法提高存储器带宽?

- (1) 增加存储器的宽度。
- (2) 采用简单的多体交叉存储器。
- (3) 独立存储体。
- (4) 避免存储体冲突。

25. 增加存储器的宽度有什么不足之处?

(1) 它会增加 CPU 和存储器之间的连接通路 (通常称为存储器总线) 的宽度, 使其实现代价提高。

(2) 由于 CPU 访问 Cache 仍然是每次访问一个字, 所以 CPU 和 Cache 之间需要有一个多路选择器, 而且这个多路选择器可能会处在关键路径上。主存宽度增加后, 用户扩充主存时的最小增量也增加了相应的倍数。在具有纠错功能的存储器中实现对一行 (一次可并行读出的数据) 中部分数据的写入比较复杂。

26. 简述 DRAM 专用交叉结构优化的三种方式。

(1) Nibble 方式 — 每次进行行访问时, DRAM 除能够给出所需的位以外, 还能给出其后的 3 位。

(2) Page 方式 — 缓冲器以 SRAM 的方式工作: 通过改变列地址, 可以随机地访问缓冲器内的任一位。这种访问可以反复进行, 直到下一次行访问或刷新。

(3) Static column 方式 — 和 Page 方式类似, 只是在列地址改变时, 无需触发列访问选通线。

第6章 输入/输出系统

1. 输入/输出系统包括哪两部分？

输入/输出系统简称 I/O 系统，它包括 I/O 设备以及 I/O 设备与处理机的连接。

2. 评价 I/O 系统性能的参数主要有哪些？

- (1) 连接特性（哪些 I/O 设备可以和计算机系统相连接）。
- (2) I/O 系统容量（I/O 系统可以容纳的 I/O 设备数）。
- (3) 响应时间。
- (3) 吞吐率。

3. 反映存储外设可靠性的参数有哪些？

- (1) 可靠性：系统从初始状态开始一直提供服务的能力。
- (2) 可用性：系统正常工作时间在连续两次正常服务间隔时间中所占的比率。
- (3) 可信性：服务的质量，即在多大程度上可以合理地认为服务是可靠的。

4. 通常用哪两个特征来区分磁盘阵列？

大多数磁盘阵列的组成可以由以下两个特征来区分：

- (1) 数据交叉存放的粒度。
- (2) 冗余数据的计算方法以及在磁盘阵列中的存放方式。

5. 在磁盘阵列中设置冗余需要解决哪两个问题？

- (1) 如何计算冗余信息。当今的磁盘阵列大多都是采用奇偶校验码，但也有采用汉明码或 Reed-Solomon 码的。
- (2) 如何把冗余信息分布到磁盘阵列中的各个盘。

6. 盘阵列有哪些分级？

- (1) RAID0：亦称数据分块，即把数据分布在多个盘上，无冗余信息。
- (2) RAID1：镜像盘，使用双备份磁盘。
- (3) RAID2：位交叉式海明编码阵列。
- (4) RAID3：位交叉奇偶校验盘阵列。即数据以位或字节交叉的方式存于各盘，冗余的奇偶校验信息存储在一台专用盘上。
- (5) RAID4：专用奇偶校验独立存取盘阵列。即数据以块（块大小可变）交叉的方式存于各盘，冗余的奇偶校验信息存在一台专用盘上。
- (6) RAID5：块交叉分布式奇偶校验盘阵列。即数据以块交叉的方式存于各盘，冗余的奇偶校验信息均匀地分布在所有磁盘上。
- (7) RAID6：双维奇偶校验独立存取盘阵列。可容忍双盘出错。
- (8) RAID7：是采用 Cache 和异步技术的 RAID6，使响应速度和传输速率有了较大提高。

7. 简述 RAID10 与 RAID0+1 的区别。

RAID10 又称为 RAID1+0，RAID01 又称为 RAID0+1，它们都是 RAID0 与 RAID1 相结合的结果，其区别在于先做什么，即先镜像还是先按条带存放。RAID1+0 是先进进行镜像 (RAID1)，

然后再进行条带存放 (RAID0)。RAID0+1 是先进行条带存放 (RAID0)，然后再进行镜像 (RAID1)，即组内按条带存放，组之间做镜像。

8. 实现盘阵列的方式有哪几种？

实现盘阵列的方式主要有三种：

(1) 软件方式：即阵列管理软件由主机来实现。其优点是成本低，缺点是要过多地占用主机时间，并且带宽指标上不去。

(2) 阵列卡方式：即把 RAID 管理软件固化在 I/O 控制卡上，从而可不占用主机时间，一般用于工作站和 PC 机。

(3) 子系统方式：这是一种基于通用接口总线的开放式平台，可用于各种主机平台和网络系统。

9. 磁盘阵列技术研究的主要热点问题有哪几个方面？

(1) 新型阵列体系结构。

(2) RAID 结构与其所记录文件特性的关系。

(3) 在 RAID 冗余设计中，综合平衡性能、可靠性和开销的问题。

(4) 超大型磁盘阵列在物理上如何构造和连结的问题等。

10. 按用途分类，总线可分为哪两类？它们各有何特点？

(1) CPU 存储器总线。CPU 存储器总线比较短，通常具有较高的速度，并且要和存储器系统的速度匹配来优化带宽。

(2) I/O 总线。I/O 总线要连接许多不同类型、不同带宽的设备，因而比较长，并且应遵循总线标准。

11. 按设备定时方式分类，总线可分为哪两类？它们各有何特点？

(1) 同步总线。同步总线上所有设备通过统一的总线系统时钟进行同步。同步总线成本低，因为它不需要设备之间互相确定时序的逻辑。但是同步总线也有缺点，总线操作必须以相同的速度运行。

(2) 异步总线。异步总线上的设备之间没有统一的系统时钟，设备自己内部定时。设备之间的信息传送用总线发送器和接收器控制。异步总线容易适应更广泛的设备类型，扩充总线时不用担心时钟时序和时钟同步问题。但在传输时，异步总线需要额外的同步开销。

12. 通道分为哪三种类型？它们分别适合为哪种外围设备服务？

(1) 字节多路通道：一种简单的共享通道，主要为多台低速或中速的外围设备服务。

(2) 数组多路通道：适于为高速设备服务。

(3) 选择通道：为多台高速外围设备服务。

13. 通道的主要功能有哪些？

(1) 接收 CPU 发来的 I/O 指令，根据指令要求选择一台指定的外围设备与通道相连接。

(2) 执行 CPU 为通道组织的通道程序，从主存中取出通道指令，对通道指令进行译码，并根据需要向被选中的设备控制器发出各种操作命令。

(3) 给出外围设备的有关地址，即进行读 / 写操作的数据所在的位置。

- (4) 给出主存缓冲区的首地址，这个缓冲区用来暂时存放从外围设备上输入的数据，或者暂时存放将要输出到外围设备中去的数据。
- (5) 控制外围设备与主存缓冲区之间数据交换的个数，对交换的数据个数进行计数，并判断数据传送工作是否结束。
- (6) 指定传送工作结束时要进行的操作。
- (7) 检查外围设备的工作状态是正常或故障。根据需要将设备的状态信息送往主存指定单元保存。
- (8) 在数据传输过程中完成必要的格式变换。

14. 通道完成一次数据传输的主要过程？

- (1) 在用户程序中使用访管指令进入管理程序，由 CPU 通过管理程序组织一个通道程序，并启动通道。
- (2) 通道处理机执行 CPU 为它组织的通道程序，完成指定的数据 I/O 工作。
- (3) 通道程序结束后向 CPU 发中断请求。CPU 响应这个中断请求后，第二次进入操作系统，调用管理程序对 I/O 中断请求进行处理。

15. 简述字节多路通道的数据传送过程。

在字节多路通道中，通道每连接一个外围设备，只传送一个字节，然后又与另一台设备连接，并传送一个字节。

16. 简述数组多路通道的数据传送过程。

数组多路通道每连接一台高速设备，一般传送一个数据块，传送完成后，又与另一台高速设备连接，再传送一个数据块。

17. 简述选择通道的数据传送过程。

在选择通道中，通道每连接一个外围设备，就把这个设备的 n 个字节全部传送完成，然后再与另一台设备相连接。

18. 如果使用物理地址进行 DMA 传输，存在什么问题？解决方法有哪些？

存在的问题：

- (1) 对于超过一页的数据缓冲区，由于缓冲区使用的页面在物理存储器中不一定是连续的，所以传输可能会发生问题。
- (2) 如果 DMA 正在存储器和缓冲区之间传输数据时，操作系统从存储器中移出（或重定位）一些页面，那么，DMA 将会在存储器中错误的物理页面上进行数据传输。

解决方法：一种方法是使操作系统在 I/O 的传输过程中确保 DMA 设备所访问的页面都位于物理存储器中，这些页面被称为是钉（pinned）在了主存中。另一种方法是采用“虚拟 DMA”技术，它允许 DMA 设备直接使用虚拟地址，并在 DMA 期间由硬件将虚拟地址转换为物理地址。

第 7 章 互连网络

1. 互连网络中有哪两种定时方式？

有同步和异步两种。

同步系统使用一个统一的时钟。而异步系统没有统一的时钟，系统中的各个处理机都是独立地工作。

2. 互连网络中有哪两种交换方法？

有线路交换和分组交换两种。

在线路交换中，源结点和目的结点之间的物理通路在整个数据传送期间一直保持连接。在分组交换中，是把信息分割成许多组（又称为包），将它们分别送入互连网络。这些数据包可以通过不同的路径传送，到目的结点后再拼合出原来的数据。在分组交换中，结点之间不存在固定连接的物理通路。

3. 互连网络中有哪两种控制策略？

有集中式和分散式两种。

集中控制方式中，有一个全局的控制器接收所有的通信请求，并由它设置互连网络的开关连接。分散控制方式中，不存在全局的控制器，通信请求的处理和开关的设置由互连网络分散地进行。

4. 按照拓扑结构，互连网络分为哪两类？

有静态和动态两种。

静态拓扑结构在各结点之间有专用的连接通路，且在运行过程中不能改变。动态拓扑结构中，可根据需要设置互连网络中的开关，从而对结点之间的连接通路进行重新组合，实现所要求的通信模式。

5. 写出几种常用的基本互连函数。

(1) 交换函数：实现二进制地址编码中第 k 位互反的输入端与输出端之间的连接。

(2) 均匀洗牌函数 (shuffle)：将输入端分成数目相等的两半，前一半和后一半按类似均匀混洗扑克牌的方式交叉地连接到输出端（输出端相当于混洗的结果）。

(3) 蝶式互连函数：把输入端的二进制编号的最高位与最低位互换位置，便得到了输出端的编号

(4) 反位序函数是将输入端二进制编号的位序颠倒过来求得相应输出端的编号。

(5) PM2I 函数：是一种移数函数，它是将各输入端都循环移动一定的位置连到输出端。

6. 互连网络的主要特性参数有哪几个？

(1) 网络规模：网络中结点的个数。它表示该网络所能连接的部件的数量。

(2) 结点度：与结点相连接的边数（通道数），包括入度和出度。

(3) 距离：对于网络中的任意两个结点，从一个结点出发到另一个结点终止所需要跨越的边数的最小值。

(4) 网络直径：网络中任意两个结点之间距离的最大值。网络直径应当尽可能地小。

(5) 结点之间的线长：两个结点之间连线的长度，用米、千米等表示。

(6) 等分宽度：当某一网络被切成相等的两半时，沿切口的边数（通道数）的最小值称为通道等分宽度。

(7) 对称性：从任何结点看到的拓扑结构都是相同的网络称为对称网络。

7. 动态连接网络有哪几种？

(1) 总线：价格较低、带宽较窄、容易产生故障。

(2) 交叉网络：价格昂贵、带宽和互连特性最好。

(3) 多级网络：采用模块结构，扩展性较好；其时延随网络的级数而上升。

第 8 章 多处理机

1. 根据多处理机系统中处理器个数的多少，可把现有的 MIMD 机器分为哪两类？

根据多处理机系统中处理器个数的多少，把现有的 MIMD 机器分为：

第一类为集中式共享存储器结构；

第二类为分布式存储器结构；

每一类代表了一种存储器的结构和互连策略。

2. 在分布式存储器结构的机器中，将存储器分布到各结点有什么好处？

将存储器分布到各结点有两个好处：

第一，如果大多数的访问是针对本结点的局部存储器，则可降低对存储器和互连网络的带宽要求。

第二，对局部存储器的访问延迟低。分布式存储器结构最主要的缺点是处理器之间的通信较为复杂，且各处理器之间访问延迟较大。

3. 在分布式存储器结构的机器中，目前有哪两种存储器地址空间的组织方案？

(1) 第一种方案：物理上分离的多个存储器作为一个逻辑上共享的存储空间进行编址。

(2) 第二种方案：整个地址空间由多个独立的地址空间构成，它们在逻辑上也是独立的，远程的处理器不能对其直接寻址。

4. 在分布式存储器结构的机器中，对应于两种地址空间的组织方案，分别有哪两种通信机制？它们是怎么实现的？

(1) 共享地址空间的机器：可利用 load 和 store 指令中的地址隐含地进行数据通信，因而可称为共享存储器机器。

(2) 多个地址空间的机器：根据简单的网络协议，通过传递消息来请求某些服务或传输数据，从而完成通信。因而这种机器常称为消息传递机器。

5. 用哪三个关键的性能指标来衡量通信机制的性能？

(1) 通信带宽：理想状态下的通信带宽受限于处理器、存储器和互连网络的带宽。

(2) 通信延迟：通信延迟 = 发送开销 + 跨越时间 + 传输延迟 + 接收开销。

(3) 通信延迟的隐藏：如何才能较好地将通信和计算或多次通信之间重叠起来。

6. 共享存储器通信机制主要有哪些优点？

(1) 与常用的集中式多处理机使用的通信机制兼容。

(2) 当处理器通信方式复杂或程序执行动态变化时，易于编程；同时在简化编译器设计方面占有优势。

(3) 当通信数据较小时，通信开销较低，带宽利用较好。

(4) 通过硬件控制的 Cache 减少了远程通信的频度，减少了通信延迟以及对共享数据的访问冲突。

7. 消息传递通信机制主要有哪些优点？

(1) 硬件较简单。

(2) 通信是显式的，从而引起编程者和编译程序的注意，着重处理开销大的通信。

8. 在消息传递通信机制的硬件上怎样支持共享存储器？

所有对共享存储器的访问均要求操作系统提供地址转换和存储保护功能，即将存储器访问转换为消息的发送和接收。

9. 并行处理面临着哪两个重要挑战？

第一个是程序中有限的并行性。有限的并行性使机器要达到高的加速比十分困难。

第二个是相对较高的通信开销。