

实例分析

- 一个字节多路通道连接有6台设备，它们的数据传输率如下表所示。

设备名称	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆
数据传输速率 (B/ms)	50	50	40	25	25	10

- 计算该通道的实际工作流量。
- 若通道的最大流量等于实际工作流量，求通道的工作周期 $T_S + T_D$ 。

实例分析

设某个字节多路通道的设备选择时间 T_S 为 $9.8\mu s$ ，传送一个字节的数据所需的时间 T_D 为 $0.2\mu s$ 。若某种低速外设每隔 $500\mu s$ 发出一次传送请求，那么，该通道最多可接多少台这种外设？

解：字节多路通道的最大流量为：
$$f_{\max\text{-byte}} = \frac{1}{T_S + T_D}$$

字节多路通道的实际流量为：
$$f_{\text{byte}} = \sum_{i=1}^p f_i$$

其中， p 为通道连接的外设台数， f_i 为外设 i 的数据传输速率。因为连接的是同样的外设，所以 $f_1 = f_2 = \dots = f_p = f$ ，故有 $f_{\text{byte}} = pf$ 。

通道流量匹配的要求有： $f_{\max\text{-byte}} \geq f_{\text{byte}}$

即有：
$$\frac{1}{T_S + T_D} \geq pf \quad ; \quad \text{可得：} \quad p \leq \frac{1}{(T_S + T_D)f}$$

已知 $T_S = 9.8\mu s$ ， $T_D = 0.2\mu s$ ， $1/f = 500\mu s$ ，可求出通道最多可连接的设备台数为：

$$p \leq \frac{1}{(T_S + T_D)f} = \frac{500\mu s}{(9.8 + 0.2)\mu s} = 50$$

8/19

http://www.mhhe.com 计算机体系结构

例题：一个具有32台处理机的系统，对远程存储器访问时间是2000ns。除了通信以外，假设计算中的访问均命中局部存储器。当发出一个远程请求时，本地处理机挂起。处理机的时钟周期时间是10ns，假设指令基本的CPI为1.0（设所有访存均命中Cache）。对于下述两种情况：没有远程访问；0.5%的指令需要远程访问。

试问前者比后者快多少？

解：已知远程访问率 $p = 0.5\%$,

远程访问时间 $t = 2000\text{ns}$, 时钟周期 $T = 10\text{ns}$

远程访问开销 $C = t/T = 2000\text{ns}/10\text{ns} = 200$ (时钟周期数)

有 0.5% 远程访问的机器的实际 CPI_2 为：

$CPI_2 = CPI_1 + p \times C = 1.0 + 0.5\% \times 200 = 2.0$

只有局部访问的机器的基本 $CPI_1 = 1.0$

$CPI_2 / CPI_1 = 2.0 / 1.0 = 2$ (倍)

因此，没有远程访问状态下的机器速度是有0.5% 远程访问的机器速度的2 倍。

https://blog.csdn.net/qq_38570633

例：某台含64块处理器的多处理机，处理器的时钟频率为2GHz,指令基本的CPI= 0.2(设所有访存均命中Cache)。除通信以外，假设其他所有访问均命中局部存储器。对远程存储器访问时间为20ns,当发出一个远程请求时，本处理器挂起。求在没有远程访问的情况下和有1%的指令需要远程访问的情况下，前者比后者快多少？

□ 解:远程访问开销为:

□ 远程访问时间时钟周期时间=20nsx2GHz=42.95个时钟周期
有1%远程访问的机器，其实际CPI为:

□ $CPI = \text{基本CPI} + \text{远程访问率} \times \text{远程访问开销} = 0.2 + 1\% \times \text{远程访问开销} \approx 0.2 + 1\% \times 4295 = 0.6295$

□ 所以，机器速度在没有远程访问情况下是有1%远程访问情况下的0.6295/0.2约3.1475 倍。

https://blog.csdn.net/qq_38570633

2、(12 分) 给定以下的假设，

- (1) 对指令 Cache 的访问占全部访问的 75%，对数据 Cache 的访问占 25%；
- (2) Cache 的命中时间为 1 个时钟周期，失效开销为 50 个时钟周期；
- (3) 在混合 Cache 中一次 load 或 store 操作访问 Cache 的命中时间都要增加 1 个时钟周期；
- (4) 32KB 的指令 Cache 的失效率为 0.39%，32KB 的数据 Cache 的失效率为 4.82%，64KB 的混合 Cache 的失效率为 1.35%；
- (5) 采用写直达策略，且有一个写缓冲器，并且忽略写缓冲器引起的等待。

要求：

- (1) 对于指令 Cache 和数据 Cache 容量均为 32 KB 的分离 Cache，计算其失效率，且计算其平均访存时间；
- (2) 计算容量为 64 KB 的混合 Cache 的平均访存时间；
- (3) 分别从失效率和平均访存时间两个方面，比较分离 Cache 与混合 Cache 的性能。

3、(15 分) 假设某台计算机的特性及性能为：

(1) 存储器总线宽度为 1 个字 (32 位)，送地址需要 4 个时钟周期，每个字的访问时间为 24 个时钟周期，传送 1 个字的数据需 4 个时钟周期；

(2) Cache 块大小为 1 个字时，Cache 失效率为 3%；

(3) 平均每条指令访存 1.2 次；

(4) 在不考虑 Cache 失效时，平均 CPI 为 2。

要求：

(1) 计算 Cache 失效开销，存储器的带宽以及在考虑 Cache 失效时的 CPI；

(2) 若 Cache 块大小为 2 个字时，Cache 失效率为 2%，计算相应的 CPI；

(3) 若 Cache 块大小为 2 个字时，讨论在采用 2 路多体交叉存取以及将存储器和总线宽度增加一倍时，对提高性能（用 CPI 说明）各有何作用？

、(12 分)

解：

(1) 对于分离 Cache 的总体失效率为：

$$(75\% \times 0.39\%) + (25\% \times 4.82\%) = 1.4975\%$$

平均访存时间公式可以分为指令访问和数据访问两部分：

平均访存时间 = 指令所占的百分比 \times (指令命中时间 + 指令失效率 \times 失效开销) + 数据所占的百分比 \times (数据命中时间 + 数据失效率 \times 失效开销)

$$\begin{aligned} \text{分离 Cache 的平均访存时间} &= 75\% \times (1 + 0.39\% \times 50) + 25\% \times (1 + 4.82\% \times 50) \\ &= 1.74875 \end{aligned}$$

$$\begin{aligned} \text{(2) 混合 Cache 的平均访存时间} &= 75\% \times (1 + 1.35\% \times 50) + 25\% \times (1 + 1 + 1.35\% \times 50) \\ &= 1.925 \end{aligned}$$

(3) 从失效率方面来看，分离 Cache 为 1.4975%，其大于混合 Cache 的失效率 1.35%，混合 Cache 的性能优于分离 Cache。

但从平均访存时间来看，分离 Cache 为 1.74875 个时钟周期，其小于混合 Cache 的 1.925 个时钟周期，分离 Cache 的性能优于混合 Cache。

因此，尽管分离 Cache 的实际失效率比混合 Cache 的高，但分离 Cache 提供了两个端口，消除了结构冲突，其平均访存时间反而较低。

3、(15 分)

解：

(1) 当 Cache 块大小为一个字时，

Cache 失效开销为 $4 + 24 + 4 = 32$ 个时钟周期。

存储器的带宽为每个时钟周期 $4/32 = 1/8$ 字节。

在考虑 Cache 失效时的 CPI 为 $2 + (1.2 \times 3\% \times 32) = 3.15$ 个时钟周期。

(2) 若 Cache 块大小为 2 个字时，Cache 失效率为 2%，

相应的 CPI 为 $2 + (1.2 \times 2\% \times 2 \times 32) = 3.54$ 个时钟周期。

(3) 若 Cache 块大小为 2 个字且采用 2 路多体交叉存取时，

相应的 CPI 为 $2 + 1.2 \times 2\% \times (4 + 24 + 8) = 2.86$ 个时钟周期。

性能相对于不采用 2 路多体交叉存取，提高了 $(3.54 - 2.86) / 3.54 = 19.21\%$ 。

若 Cache 块大小为 2 个字且采用 64 位总线和存储器，不采用多体交叉时，

相应的 CPI 为 $2 + 1.2 \times 2\% \times 1 \times 32 = 2.77$ 个时钟周期。

性能相对于不采用存储器和总线宽度增加一倍，提高了 $(3.54 - 2.77) / 3.54 = 21.75\%$ 。

判断题

- (√) 执行时间不是唯一的性能指标，但它是最普遍的性能表示形式。
- (×) 根据 Moore 定律，DRAM 的周期时间是每三年降低四倍。
- (×) MIPS 是测量计算机性能的一致性指标。
- (√) 在计算机性能测量中，调和平均的含义是单位时间机器能够执行的程序数。
- (×) 可以用典型程序来设计和优化指令集。
- (×) 增加流水线的级数总可以增加流水线的性能。
- (×) 多处理机系统中的素数模低位交叉存储器可以避免所有访存冲突。
- (√) 部件的可靠性通常可以用平均无故障时间来衡量。

- (×) RISC 结构的机器性能一定要比 CISC 结构的机器性能高。
- (×) 平均每条指令的执行周期数 (CPI) 与程序无关。
- (√) CPU 性能公式中指令条数 (IC) 与指令集格式和编译器有关。
- (√) CPU 的组织在一定程度上会影响 CPU 所能达到的频率。
- (√) 解释执行比翻译执行花的时间多，但存储空间占用较少。
- (×) 计算机体系结构设计这不必关心指令集具体实现。
- (×) 当前没有任何一种指令集结构是堆栈型结构，因为它已经过时了。
- (√) 虽然结构相关会影响流水线的性能，但是我们在具体的流水线设计中仍然允许一定的结构相关存在。
- (×) 程序的时间局部性指程序即将用到的信息很可能与目前正在使用的信息在空间上相邻或者临近。
- (×) 程序的空间局部性指程序即将用到的信息很可能就是目前正在使用的信息。
- (√) Amdahl 定律揭示的性能递减规则说明如果仅仅对计算机中的一部分做性能改进，则改进越多，系统获得的效果越小。
- (×) Amdahl 定律中“可改进比例”指可改进部分在改进系统计算时间中所占的比例。
- (√) Amdahl 定律中“部件加速比”指可改进部分改进以后性能的提高。
- (√) 传统机器级的机器语言是该机的指令集。
- (×) 由汇编语言写成的程序叫做汇编程序。
- (√) 机器功能的软件和硬件实现在逻辑上是等效的，但性能价格比是不等效的。
- (√) 集成电路基片成本和基片面积有一定比例关系。
- (√) 芯片研发费用指芯片毛利的一部分。
- (×) 集成电路芯片的成本主要取决于芯片生产数目。
- (√) 响应时间必须通过运行“真实程序”获得。

- (×) 通过操作系统的多进程技术可以完全回避 I/O 处理时间的问题。因为当一个进程在等待 I/O 处理的时候，另外的一些进程可以在 CPU 上运行。
- (×) 在 Cache 中，只要增加块容量，一定可以减小失效率。
- (√) 虚拟 Cache 中，CPU 使用虚拟地址访问 Cache。
- (×) 虚拟存储器页调度采用 LRU 算法的缺点之一是使 CPU 时钟频率下降。
- (√) 可以说向后兼容是系列机的根本特征。
- (×) 软硬件功能是完全等效的。
- (×) 机器工作状态的定义和切换对机器语言程序设计者来说是透明的。

- (×) 基准程序能够完全预测一个程序在计算机上的运行性能。
- (√) 随着时间的推移, 计算机系统的成本会逐渐降低。
- (×) 现代 CPU 中已经没有堆栈型和累加器型的指令集结构。
- (√) 编译器对应用程序的优化编译会增加控制指令在程序中所占的比例。
- (×) Load/Store 型机器体系结构设计一定要遵循指令集结构的规整性要求。
- (×) 流水线的效率在满负荷运行时可达 1。
- (√) 用时空图上 n 个任务所占的时空区与 m 个段总的时空区之比可计算出流水线的效率。
- (×) 由于流水线的最大加速比等于流水线深度, 所以增加流水段数总可以增大流水线加速比。
- (√) 流水线深度受限于流水线的延迟和额外开销。
- (√) 编译器可以通过重新排列代码的顺序来消除相关引起的暂停。
- (√) 多级存储层次是利用程序局部性原理来设计的。
- (√) “Cache – 主存”层次: 弥补主存速度的不足。
- (√) “主存 – 辅存”层次: 弥补主存容量的不足。
- (√) 写调块策略是用于写操作失效时的策略。
- (√) 写合并是提高写缓冲利用率的技术。
- (√) 相联度越高, 冲突失效就越少。
- (×) 强制性失效和容量失效也受相联度的影响。
- (×) 容量失效却随着容量的增加而增加。
- (√) 2:1 的 Cache 经验规则说明容量为 N 的直接映象 Cache 的失效率约等于大小为 $N/2$ 的两路组相联 Cache 的失效率。
- (√) 一些降低失效率的方法会增加命中时间或失效开销。
- (×) 具有越低失效率的计算机系统性能越高。
- (×) 具有越低平均访存时间的系统性能越高。
- (×) 具有越低失效率的存储系统性能越高。
- (√) 具有越低平均访存时间的存储系统性能越高。
- (×) Victim Cache 是位于 CPU 和 Cache 间的又一级 Cache。
- (×) 伪相联 cache 取直接映象及组相联两者的优点, 命中时间小, 失效开销低。
- (√) 伪相联 cache 具有快速命中与慢速命中两种命中时间。
- (×) 预取必须和正常访存操作并行才有意义。
- (√) 预取必须和正常指令的执行并行才有意义。
- (√) 数据对存储位置的限制比指令的少, 因此更便于编译器优化。
- (√) Cache 中的写缓冲器导致对存储器访问的复杂化。
- (×) 写回法 Cache 中不必使用写缓冲。
- (×) 评价第二级 Cache 时, 应使用局部失效率这个指标。
- (√) 为减少平均访存时间, 可以让容量较小的第一级 Cache 采用较小的块, 而让容量较大的第二级 Cache 采用较大的块。
- (√) 预取只有在处理器能继续执行其它指令的同时进行才有意义。

- (√) 硬件预取通常需要非阻塞 cache 的支持。
- (√) Cache 命中时间往往会直接影响到处理器的时钟频率。
- (√) 采用容量小、结构简单的 Cache 会减小 cache 的命中时间。
- (√) 写操作流水化会减小 cache 的命中时间。

- (√) TLB 是页表转换查找缓冲器。
- (√) TLB 中的内容是页表部分内容的副本。
- (×) 在共享存储器上支持消息传递比在消息传递的硬件上支持共享存储器困难得多。
- (√) 流水线吞吐率是指单位时间内流水线所完成的任务数。
- (×) 流水线可能达到其最大吞吐率。
- (×) 流水线加速比是指流水线最大润如率和实际吞吐率之比。
- (√) Cache 失效中必定包含强制性失效。
- (×) Cache 失效中必定包含容量失效。
- (×) Cache 失效中必定包含冲突失效。
- (√) 组相联或直接映象 Cache 中才可能存在冲突失效。
- (√) 支持“失效下命中”的 cache 是非阻塞 Cache。
- (×) 虚存系统所用的 cache 称为虚拟 Cache。
- (√) 有统一的时钟协调各个设备操作的总线是同步总线。
- (×) 硬件在预取时，如果出现虚地址故障或违反保护权限，就会发生异常。
- (×) 多处理机系统由多个不同类型的处理机组成。
- (√) 分布式共享多处理机是存储器分布到各个处理器上的多处理机系统。
- (√) 由多个同种类型组成的处理机称为同构型多处理机。
- (√) 同步消息传递机制中，处理器一个请求发出后一直要等到收到应答结果才能继续运行。

简答题

填空题

计算机体系结构试题库

填空题 (100 题)

当代计算机体系结构的概念包括（指令集结构）、（计算机组成）和（计算机实现）三个方面的内容。

计算机部件的平均出售价是（部件开销）、（直接开销）和（毛利）三者之和。

在一个字中，两种表示字节顺序的习惯是（高端）和（低端）。

通常根据 CPU 内部状态，可以将指令集结构分为（堆栈型）、（累加器型）和（通用寄存器型）三种类型。

在指令流水线中，解决控制相关的方法主要有：（冻结或排空流水线）、（预测发生）、（预测不发生）和（调度分支延迟）。

I/O 性能评价的指标主要包括：设备类型、设备数量、（响应时间）和（吞吐量）。

提高向量处理机性能的主要方法有：链接、（重叠执行）和（多个向量载入储存部件）。

一般并行性包含（并行）和（并发）两个方面。

开发并行性的主要途径有：（时间重叠）、（资源重复）和（资源共享）。

指令内部的并行属于（细）粒度并行。

流水线的数据相关有（RAW）、（WAW）、（WAR）三种类型。

通用寄存器型指令集结构按其指令中的操作数个数和操作数的存储单元可以分为（RR）、（RM）、（MM）三种类型。

根据 CPU 性能公式，程序的执行时间等于（ IC ）、（ CPI ）及（ T ）三者的乘积。
DLX 流水线可以分为（ IF ）、（ ID ）、（ EX ）、（ MEM ）、（ WB ）五个操作功能段。
在存储器层次结构中，Cache 离 CPU（最近），而外存离 CPU 最远。
一般来说，按照 CPU 内部操作数的存储方式，可以将机器（指令集结构）分为：（堆栈型）、（累加器型）和（通用寄存器型）三种类型。
单机和多机并行性发展的技术途径有：（资源共享）、（资源重复）和（时间重叠）。
存储器层次结构设计技术的基本依据是程序（访问的局部性原理）。
在计算机体系结构设计中，软硬件功能分配取决于（性能价格比）。
从主存的角度来看，“Cache—主存”层次的目的是为了（提高速度），而“主存—辅存”层次的目的是为了（扩大容量）。

程序循环是用（转移指令）来实现，而微程序循环是用（微指令地址转移测试方法）来实现的。

计算机组成指的是计算机系统结构的逻辑实现，计算机实现指的是计算机组成的物理实现
存储程序计算机以运算器为中心、所有部件的操作都由控制器集中控制。
指令集结构的正交特性是指令集的三个主要元素操作、数据类型和寻址方式两两在指令集结构中独立无关。

通道可分为三类：字节多路通道，选择通道，数组多路通道。

Cache 的调度算法通常有预取法和按需取进法两种。

Cache 失效可以分为 强制性失效 、 容量失效 和 冲突失效 三种。

地址映象方法有多种，其中的直接相联硬件开销最小，全相联的冲突概率最小。

根据存储映象算法的不同，虚拟存储器主要有 段式、页式和段页式 三种映象方式。

流水技术按处理的级别可分为部件级、处理机级和系统级。

通常，在进行指令集格式设计时，有（固定长度编码）、（可变长编码）和（混合编码）三种设计方法。

综合考虑不同的存储器实现技术，我们会发现：速度越快，每位价格就（越高）；容量越大，每位价格就（越低）；容量越大，速度（越慢）。

磁盘的每一磁道分成若干扇区，它是磁盘进行存储分配的物理基本单元，它们之间留有（不用的间隙）。

系列机的软件兼容主要包括（向前兼容）、（向后兼容）、（向下兼容）、（向上兼容）四种类型的兼容。

Amdahl 定律表明系统的加速比依赖于（被加速部分在系统中所占的比例）和（对被加速部分的性能提高程度）两个因素。

在大多数指令集结构的功能设计中必须考虑支持的三种类型的指令是（数据传输指令）、（算术和逻辑运算指令）和（控制指令）。

在指令系统设计中，表示寻址方式有（将寻址方式编码与操作码中）和（用地址描述符表示寻址方式）两种方法。

通常，在进行指令集格式设计时，有（定长）、（变长）和（混合）三种设计方法。

在大多数指令集结构的功能设计中必须考虑支持的三种类型的指令是（算术和逻辑运算）、（数据传输）和（控制）。

在指令系统设计中,操作数类型的表示主要有(由操作码编码)和(附上由硬件解释的标记)两种方法。

在“Cache-主存”层次中,主存的更新算法有两种:(写回法)和(写直达法)。

在“Cache-主存”层次中,cache 写失效时采用的两种调块策略有:(按写分配)和(绕写法)。

设计 I/O 系统的三个标准是(性能)、(价格)和(容量)。

互连网络根据工作行为可分为两类,一种是(动态网络),一种是(静态网络)。

DLX 流水线可以分为(取指)、(译码)、(执行)、(访存)、(写回)五个操作功能段。

基本 DLX 流水线中,假设分支指令需要 4 个时钟周期,其它指令需要 5 个时钟周期,分支指令占总指令数的 12%,问 $CPI=4.88$,若把 ALU 指令的写回提前到 MEM 段,ALU 指令占总指令数的 44%,则 $CPI=4.44$ 。

消除瓶颈的两种方法为细分瓶颈段和重复设置瓶颈段。

当流水线中数据和指令存在同一存储器中时,访存指令会引起存储器访问冲突,这种冲突是因为结构相关引起的。

多级存储层次是利用程序局部性原理来设计的。

评价 cache 系统速度快慢的指标是平均访问时间。

CPU 时间能够评价 cache 系统对整个 CPU 性能的影响。

响应时间是指从事件开始到结束之间的时间。

吞吐率指在单位时间内所能完成的工作量(任务)。

用户以响应时间为标准评价计算机性能。

多道程序系统以吞吐率为标准评价计算机性能。

流水线各个功能段所需时间应尽量相等。

Cache 并行查找的两种实现方法是:利用相联存储器和利用单体多字存储器+比较器。

假设某程序中 Load 指令占 26%,Store 指令占 9%,则写操作在所有访存操作中所占的比例为 7%,写操作在访问数据 Cache 操作中所占的比例为 25%。

改进 Cache 的性能的三种途径是降低失效率、减少失效开销、减少 Cache 命中时间。

减小强制性失效的方法有:增加块大小,预取。

减小容量失效方法是增加容量。

减小冲突失效的方法是提高相联度。

容量为 128KB 的 8 路组相联 Cache 命中时间为 1.14ns,失效率为 0.6%,失效开销为 50ns,则其平均访存时间为 1.44。

伪相联 cache 相对于组相联 cache 的缺点是:具有多种命中时间。

两级 cache 的应使第一级 Cache 容量小,速度快,使第二级 Cache 容量大。

主存的主要性能指标是延迟和带宽。

通信延迟 = 发送开销 + 跨越时间 + 传输延迟 + 接收开销。

流水线中解决数据相关的技术有定向技术、暂停技术、编译器调度。

Cache 一致性协议是维护多个处理器一致性的协议。