《计算机系统基础》分析应用题

第五-八章作业

学号	5120203245	班级	卓软 2001	姓名	肖尧

1、CPU 执行指令的过程中，其他部件在做什么？（第五章）

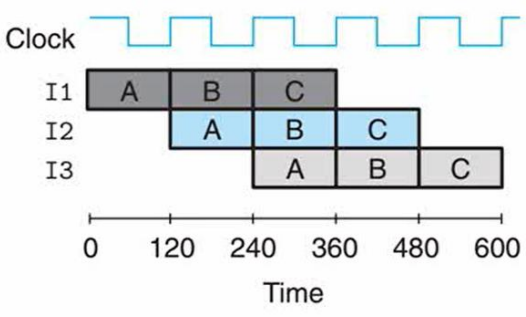
在 CPU 执行指令的过程中，其他部件也会根据指令的要求配合 CPU 完成具体的内容（读写，运算等），或者进行指令的执行控制。

冯诺依曼计算机程序的执行便是取指令，译码，取数并执行，送结果的过程，而整个执行过程的控制是由 CU 通过对指令进行译码后送出控制信号来进行的。如果指令中包含对存储器或 I/O 设备的访问，由 CPU 通过总线将要访问的地址和操作等信息送到相应设备控制器，在控制信号的控制下，相应 I/O 设备进行相应做出对应操作。

例如，若不采用 cache，则每次指令执行前，都要通过向总线发出主存地址和读取命令来控制存储器取指令；若当前执行的是寄存器定点加法指令，则 CU 控制定点运算器进行动作.....

2、流水线深度越深，时钟频率就越高，对吗？请分析回答。（第五章）

在一定深度上是的（除非寄存器传输时间已经超过本身的时钟周期，这时由于流水段之间的信息传输的瓶颈导致时钟周期已经



不能继续减小）。

增加流水线深度，导致每个流水段内的操作变得很简单，因此每个阶段的耗时就很少，也就缩短了时钟周期，提高了时钟频率。然而，这种分析方法并没有考虑到来自寄存器之间传送数据的时间消耗（如下图所示）。当这种额外消耗的比例达到 50%时，再增加流水线的深度旧没有意义了。此外，由于流水线优化和存储器冲突处理的控制逻辑将随流水线深度的加深而大量增多，可能导致用于流水线之间的逻辑比流水段本身的控制逻辑更复杂。

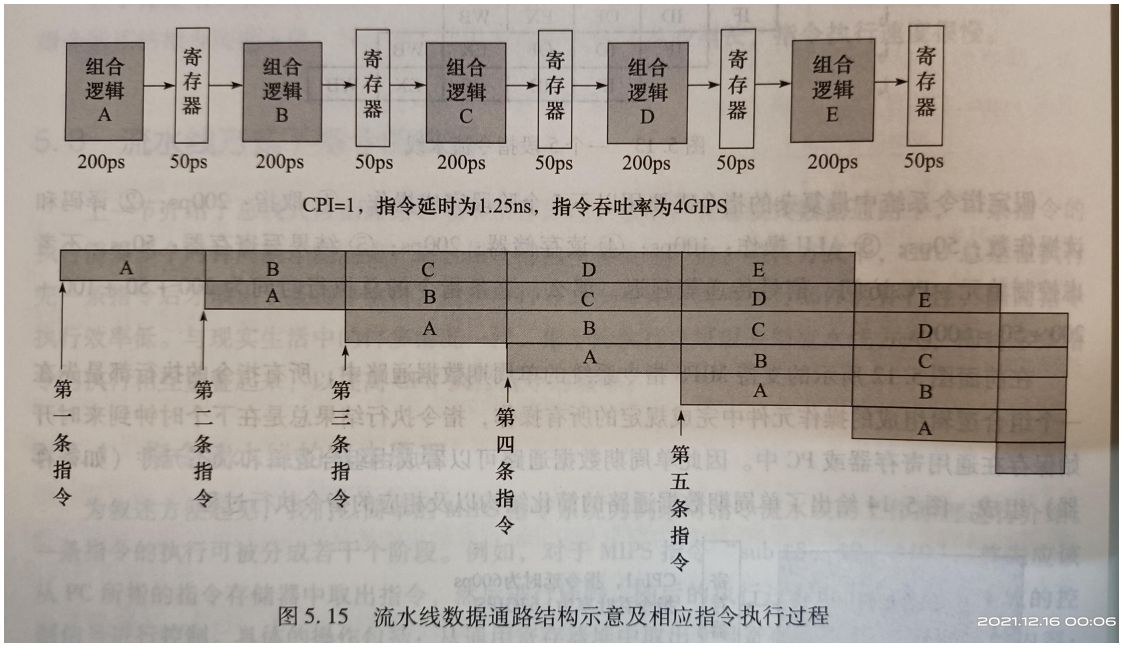


图 5.15 流水线数据通路结构示意及相应指令执行过程

2021.12.16 00:06

CPU 单核性能为什么难以提升?

AMD 和 intel 在进行核战，老是在堆核。可是，单核提升对我们来说不是更有用吗，为啥单核不容易做到领先，不断更新呢?

关注问题 写回答 邀请回答 好问题 27 1 条评论 分享 ··· 收起

其实，当时已经基本垄断了桌面 CPU 市场的 Intel 更是夸下了海口，表示奔腾 4 所使用的 CPU 结构可以做到 10GHz，但最终却失败了。奔腾 4 的主频为什么没能超过 3.8GHz 的障碍呢？这是因为时钟频率的提高必然会要求电压与功率的增高，因此达到了所谓的“energy wall”。[1]

如果想要提升 CPU 主频，无论是下面这两个操作都会增加功耗，带来耗电和散

[1] https://www.zhihu.com/question/365639711

热的问题。
增加密度：同样的面积里面，多放一些晶体管；提升主频：让晶体管“打开”和“关闭”得更快一点。
在 CPU 里面，能够放下的晶体管数量和晶体管的“开关”频率也都是有限的。
一个 CPU 的功率，可以用这样一个公式来表示：
功耗 = 1/2 × 负载电容 × 电压的平方 × 开关频率 × 晶体管数量
功耗增加太多，就会导致 CPU 散热跟不上，这时，我们就需要降低电压。这里有一点非常关键，在整个功耗的公式里面，功耗和电压的平方是成正比的。这意味着电压下降到原来的 1/5，整个的功耗会变成原来的 1/25。事实上，从 5MHz 主频的 8086 到 5GHz 主频的 Intel i9，CPU 的电压已经从 5V 左右下降到了 1V 左右。这也是为什么我们 CPU 的主频提升了 1000 倍，但是功耗只增长了 40 倍。

4、怎么知道要找的指令或数据不在内存中？（第六章）

要找的指令或数据不在内存中即在取某条指令或存取某个数据时发生了缺页情况。是否缺页主要是通过查看对应页表中的有效位是否为0来判断。

其过程为：根据要找的指令或操作数的地址高位，确定所访问的虚页号，以虚页号作为索引值，找到对应的页表项，每个页表项中都有一个有效位，若为0，则表示该页（即指令或数据所在的页）不在内存中，发生了缺页异常。

5.在层次化存储结构中，“cache→主存”“主存→磁盘”这两个层次有哪些是相似的，哪些是不同的，请比较做答。（第六章）

根据cache是主存的缓存这一规定，其实可以认为在层次化存储结构中，任何x层储存设备都是x+1层的缓存，可以认为主存也是硬盘的缓存。

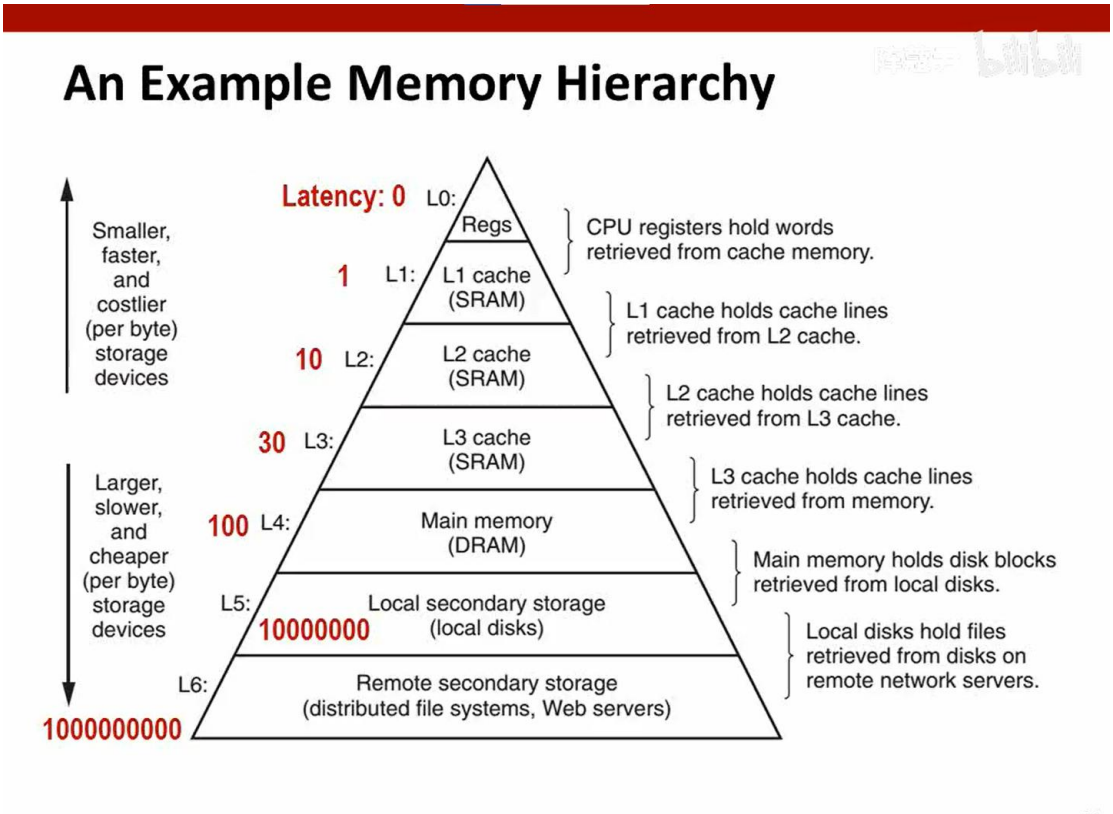
其相同点在于：①都是根据程序访问局部性的特点，将一块相邻的局部信息从慢速存储器复制到快速存储器；②都必须考虑映射问题（CT+CI+CO，页表...）③快速存储器找不到时要在慢速存储器中进行一整块调入④快速存储器满后要考虑替换。

由于其访问速度的悬殊性，所以存在不同点：①映射方式不同：由于外存的速度过慢，且内存相比cache大了不少，避免反复与磁盘进行读写操作，因此需要一次性尽量多的读取数据，因此采用全相联映射。而由于内存速度不至于那么慢且cache昂贵容量小，因此一般采用组相联映射。②写策略不同：同理由于速度的差异性，cache→主存一般write through与write back都可以使用，而主存→磁盘则基本采用write back减少对磁盘的读写。③目的不同。cache主要是为了加快CPU访问信息的速度，而虚拟存储机制则是为了营造一种每个程序独占内存的假象而方便程序员写程序。但也不能否认这种机制减少了对硬盘的访问，加快了速度。④位置不同。显而易见，不再阐述。

6. 请阐述缓存思想在计算机存储体系中哪些层次用到了缓存思想，用到以后对计算机系统性能高的提高体现在什么地方？请描述你了解的缓存思想的实现技术有哪些？我们常常提及的缓存指的是什么？有哪些类型？

如果说cache是对内存的缓存，那么内存也可以理解为对硬盘的缓存，其由虚拟存储器的页表进行映射，根据程序局部性的特点，一块相邻

的局部信息从硬盘的加载到速度更快的内存。如下图所示，根据 cache 是主存的缓存这一规定，其实可以认为在层次化存储结构中，任何 L 层储存设备都是 L+1 层的缓存。



[2]

缓存的实现需要考虑如何进行映射（根据容量，速度等因素综合考虑采取直接映射，组相联映射或全相联映射），写策略（更安全但耗时明显增多的 write through 或与之相反的 write back 策略）。

我们常常提及的缓存指的是 Cache（图中的 L1-L3。L1 一般为 CPU 专有，不在多个 CPU 中共享。L2 cache 一般是多个 CPU 共享的，也可能装在主板上。L1 cache 还可能分为 instruction cache, data cache. 这样 CPU 能同时取指令和数据。），根据不同映射方式可以分为直接映射高速缓存（E=1，即每一组仅一行，若任需要占该组的位置则只

[2] CMU CSAPP 课堂教学课件

能进行替换而不能装入该组其余行，导致数据频繁调进调出命中率低），组相联高速缓存（1<E<C/B，组中包含若干行，非常适合Cache-Memory这种均衡性缓存），全相联高速缓存（E=C/B，此时S=1，装入任意行，因此进行tag位对比的时间开销大，因此适合TLB这种数据量小的缓存使用）。

7. ROM和RAM都是随机存取存储器吗？分析回答（第六章）

是的。随机存取指的是区别于磁带那样，可以任意选择某个单元进行读写的存储器。虽然ROM和RAM放在一起进行分类，但其存取方式均是随机访问型，都是通过对地址进行译码，然后选择某个单元进行读写。至于为什么要分为ROM和RAM，正如其名ROM(read only memory)，指的是只能读不能写（只能出厂时写一次，当然最近的flash闪存已经突破了传统ROM不能写的限制），而RAM（random access memory）随机访问，则能读能写。

8. 一个进程不管中间是否被其他进程打断，也不管被打断几次，或在哪里被打断，它的逻辑控制流总是确定的，这样就可以保证一个进程的执行不管怎样被打断，其行为总是一致的。计算机系统主要靠什么机制实现这个能力？（第七章）

主要是由操作系统与CPU硬件提供的进程上下文切换与异常、中断处理机制来保证能够实现题目中描述的这种能力。

在进程的上下文切换时，操作系统需要完成以下三件事：①将当前进

程的现场信息保存到当前进程的系统级上下文的现场信息中（保护现场）；②恢复新进程的现场；③转移控制到新进程。

在异常、中断处理的响应过程中，CPU保存的最基本信息应该包括断点（中断处理后返回的地址，即eip）、断点处的机器状态（EFLAGS，CS，SS，ESP等）。

9、中断最初提出来是用于I/O设备与CPU之间传送数据的控制方式，随着技术的不断发展，中断的概念和用途已经超越了最初的内涵。请你描述一下当前中断的内涵，并给出中断给计算机系统（包括硬件和软件）管理和使用带来的好处。（第七章和第八章）

在CPU执行程序过程中，有两种情况会打断程序的执行，一种情况是CPU正在执行的指令出现了异常（fault,trap or abort），另一种情况是指令执行正常，但外部设备出现了特殊事件，要求CPU处理。一般把前者成为异常，后者成为中断。

I/O设备中，中断是指由外部I/O设备请求处理器进行处理的一种信号。由于I/O设备速度较慢，在CPU发出读写命令后，可将等待I/O的进程堵塞，先切换到别的进程执行。在涉及CPU设计时，必须考虑在数据通路中如何实现异常和中断处理，包括如何设置“开关中断”状态、如何判断是哪类异常和中断、怎样保存断点、如何切换到中断服务程序等。

中断机制是现代计算机系统中的基础设施之一，它在系统中起着通信

网络作用，以协调系统对各种外部事件的响应和处理。中断是实现多道程序设计的必要条件。中断是CPU对系统发生的某个事件作出的一种反应。引起中断的事件称为中断源。中断源向CPU提出处理的请求称为中断请求。发生中断时被打断程序的暂停点成为断点。CPU暂停现行程序而转为响应中断请求的过程称为中断响应。处理中断源的程序称为中断处理程序。CPU执行有关的中断处理程序称为中断处理。而返回断点的过程称为中断返回。中断的实现实行软件和硬件综合完成，硬件部分叫做硬件装置，软件部分成为软件处理程序。中断系统的应用大大提高了计算机效率。

10、概念辨识：（1）K、M、G在数据传输和存储容量中的意义是否相同。（2）I/O接口和I/O端口是同一个概念吗？（3）禁止中断和屏蔽中断是同一个概念吗？（第八章）

（1）不一样。在主存容量中，1K = 2^10,1M = 2^20,1G = 2^30。但是，在数据传输率中，因为数据传输速度与时钟频率有关，时钟频率通常以KHz,MHz,GHz为单位，故传输速率一般以Kbit/s,Mbit/s,Gbit/s来表示，K为10^3，M，G类推（注意这里是Bit，这也即通信公司提供的100M宽带服务传输速度仅约为12MB/s的原因）。在计算中为了进行简化，通常会将容量与速率共用1k=10^3这种表示法。

（2）不是。I/O接口指的是插在总线上的拓展卡或插件板，如网卡，

显卡控制器，声卡控制器等等（或者直接集成在主板上），其官方定义为“主机与外设之间传送信息的桥梁，介于主机与外设之间，进行控制信息，数据，状态信息等的存放。”当然，这些信息是存储在I/O接口的寄存器中，即I/O端口。

11、从用户程序提出I/O请求到外设完成I/O操作的大致过程是怎样的？（第八章）P276

用户程序若要实现I/O操作必须通过操作系统提供的I/O函数或I.O操作符请求I/O操作。例如，用户程序需要读一个磁盘文件中的记录时，它可以通过调用C语言标准库函数fread()（封装了read()），也可以直接调用操作系统提供的read()提出I/O请求。但最终都是通过操作系统内核提供的系统调用来实现的I/O。

每个系统调用的封装函数都会被转换为一组与具体机器架构相关的指令序列，这个指令序列中，至少有一个陷进指令，在陷进指令之前可能还有若干条传送指令用于将I/O操作的参数送入相应寄存器（如将系统调用号送如%eax中），然后操作系统根据系统调用号跳转执行系统调用服务例程。

I/O子系统工作的大致过程如下：首先，CPU在用户态执行用户进程，当CPU执行到系统调用的封装函数对应的指令序列中的陷进指令时，会从用户态陷入到内核态；转到内核态执行后，CPU根据陷进指令执行时EAX寄存器的系统调用号，悬着执行一个相应的系统调用服务例程；在系统调用服务例程的执行过程中可能需要调用具体设备的驱动

程序；在设备驱动程序执行过程中启动外设工作，外设准备好后发出中断请求，CPU 相应中断后，就调出中断服务程序执行，在中断服务程序中控制主机与设备进行具体的数据交换。