

简答题

1. 简述两级 Cache 的工作原理（2001）

两级 Cache 是一种降低失效开销的技术。通过在原有 Cache 和存储器之间增加另一级 Cache，构成两级 Cache，其中第一级 Cache 容量小，但速度与 CPU 的时钟周期相匹配；第二级 Cache 足够大，能够捕获更多本来要到主存去的访问，从而能够降低实际失效开销。

访存时先访问一级 Cache，若一级 Cache 命中，则直接获取数据；若一级 Cache 没命中，则访问二级 Cache。二级 Cache 没有命中才访问存储器。

2. 简述 TLB 工作原理（2001）

页表存在于主存中，页存放在虚拟存储器中，一次缺页处理将花费数百万个时钟周期。程序每次访存至少需要两次：一次访存以获得物理地址，一次访存才获得数据。为了加开访存速度，设置 TLB。TLB 是一个专用的高速缓冲器，用于存放近期经常使用的页表项，其内容是页表部分内容的副本。这样，进行地址变换时，一般直接查 TLB 就可以了，只有偶尔在 TLB 不命中时，才需要去访问内存的页表。TLB 也利用了局部性原理，如果访存具有局部性，则这些访存中的地址变换也具有局部性，即所有的页表项是相对簇聚的。TLB 也常称为“快表”或地址变换缓冲器。

TLB 中的项与 Cache 中的类似，也是由两部分构成：标识和数据。标识中存放的是虚地址的一部分，而数据部分中则存放物理页号、有效位、存储保护信息以及其他一些辅助信息。为了使 TLB 中的内容与页表保持一致，当修改页表中的某一项时，操作系统就必须保证 TLB 中没有该页表项的副本。

3. 什么是 I/O 和 Cache 数据一致性问题？如何解决这种数据一致性问题？（2001）

数据不一致问题有两个方面：

- 1) 存储器中可能不是 CPU 产生的最新数据，I/O 系统从存储器取出来使用的是陈旧数据。
- 2) I/O 与存储器交换数据滞后，在 Cache 中被 CPU 使用的可能就是陈旧数据。

解决措施：

- 1) 写直达 Cache 可以保证存储器和 Cache 有相同的数据。
- 2) 写回 Cache 则需要操作系统帮助进行数据检查。
- 3) 根据 I/O 使用的存储器地址来清除 Cache 相应块，确保 I/O 使用的数据不在 Cache 中。
- 4) 地址检查过程也可以使用硬件完成。

4. 简述存储器程序计算机（冯·诺依曼结构）的特点（2002）

机器以运算器为中心；采用存储程序原理；存储器是按地址访问的线性编址的空间；控制流由指令流产生；指令由操作码和地址码组成；数据以二进制编码表示，采用二进制运算。

5. 目前实现盘阵列的方式有哪三种？（2002）

软件方式：即阵列管理软件由主机来实现。优点是成本低；缺点是过多地占用主机时间并且带宽指标上不去。

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

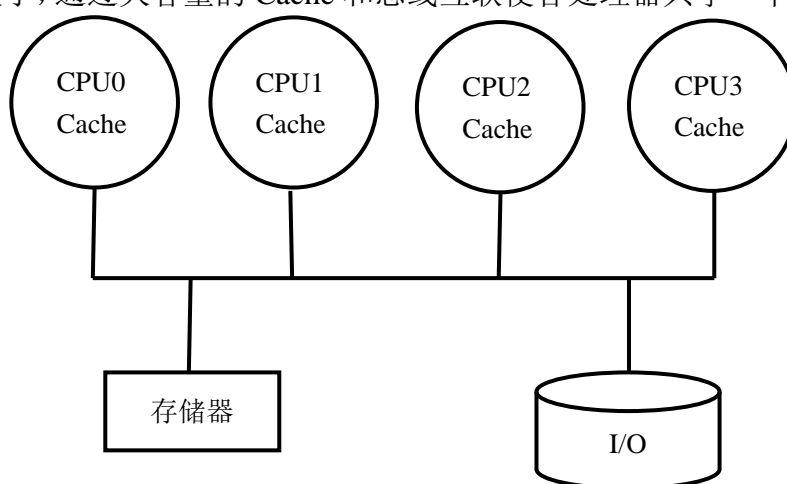
子系统方式：是一种基于通用接口总线的开放式平台，可用于各种主机平台和网络系统。

6. 简述减少 Cache 失效开销的策略。（2002）

- 1) 写缓冲及写合并：将一个数据块中的多个数据字合并一起写入存储器。或者将某些数据块合并成一个数据块再进行写。
- 2) 让读失效优于写：发生读失效时将脏块送至缓冲器，处理读失效再将脏块写入存储器。
- 3) 子块放置技术：把一个 Cache 块划分为若干小块各做标记，子块的失效开销小于整体开销。
- 4) 请求字处理技术：当 CPU 需要的数据到达后不等整个块到达 Cache 就把数据发送给 CPU。
- 5) 非阻塞 Cache 技术：发生 miss 时不停顿，继续取指执行。
- 6) 采用两级 Cache。

7. 画出共享主存多个处理器系统的概念结构，并说明其特点。（2002）

处理器数目较小，通过大容量的 Cache 和总线互联使各处理器共享一个单独集中式存储器。



8. VLIW 上性能优于向量处理器的实例。（2002）

考虑向量处理器 RAR 的指令相关，即所谓源寄存器冲突：向量寄存器不能够同时支持两个向量寄存器读，而 VLIW 可以。

ADD.D F1 F2 F3

ADD.D F4 F2 F3

ADD.D F5 F2 F3

ADD.D F6 F2 F3

9. 比较动态网络中总线、多级网络、交叉开关的特点。（2003，2008，2009）

各自特点：

- 1) 总线结构：通过共享总线把各处理器、共享存储器连接起来，在多个请求情况下，总

线仲裁分配总线的使用权。

- 2) 多级网络：每一级用了多个开关，相邻级开关之间都有固定的级间连接。
- 3) 交叉开关网络：它把 N 台处理机和 N 个存储器连接起来，网络每个交叉点是一个允许任何一台处理机和任何一个存储器连接的开关。属于无阻塞置换网络。

优缺点比较：

- 1) 总线互联最简单，造价最低，缺点是争用严重，每台处理器可用的带宽较窄，且故障率较高。
- 2) 交叉开关的硬件复杂性以 n^2 倍上升，所以其造价最为昂贵。但是交叉开关的带宽和路由性能最好。如果网络规模较小，它是一种理想的选择。
- 3) 多级网络则是两种极端的折衷。它的优点在于采用模块化结构，因此可扩展性较好。但其时延随网络级数增加而上升。另外由于增加了连线 and 开关复杂性，价格也是一种限制因素。

10. 请给出三种降低 Cache 命中时间的技术手段。(2003)

- 1) 采用容量小而结构简单的 Cache。
- 2) 采用虚拟 Cache，直接使用虚地址进行 Cache 索引。
- 3) 写操作流水化：对比标识与写入数据流水化，上一次的写入数据和当前的对比标识就可以并行。

11. 给出四种松弛一致性模型的特点及所需要的硬件支持。(2003, 2006, 2009, 2014)

松弛一致性模型为什么能提高性能？部分排序和弱排序两种模型的特点及其实现上所需要的硬件支持措施。(2007, 2010, 2012, 2015)

采用松弛 (relaxed) 一致性模型的机器提高性能的原理是什么？实现上有哪些主要的硬件支持措施？(2011)

松弛一致性允许读写操作的乱序执行，但需要同步操作保证排序原则，使得一个同步程序的表现和顺序模型一致。在保证程序正确的前提下，增加指令执行的并行性并减少时延。所以采用松弛一致性模型的机器可以提高性能。

根据消除读写顺序的内容可以划分为四类：

- 1) 完全存储排序 (TSO)：消除了 $W \rightarrow R$ ，在硬件支持上为写缓冲的读旁路等。维护写的次序，这种模型采用写缓存，并提供读的旁路机制，从而允许处理机在写的操作数被所有的别的处理机看到之前就继续进行读。
- 2) 部分存储排序 (PSO)：消除了 $W \rightarrow W$ ，在硬件支持上为写的流水线或其他并行等。允许非冲突写隐含地乱序进行。实现上，可以使写流水化或重叠，而不是强制一个操作必须在另一个操作之前结束，对同步操作仍需将写操作挂起，因为它引起写保护。
- 3) 弱排序模型 (WO)：进一步消除了 $R \rightarrow R$ ， $R \rightarrow W$ ，在硬件支持为不封锁读。
- 4) 释放一致性模型 (RC)：进一步消除了 $W \rightarrow S_A$ ， $R \rightarrow S_A$ ， $S_R \rightarrow W$ ， $S_R \rightarrow R$ ，在硬件上支持为不封锁读、旁路、无序写等。这种模型区分同步操作中的访问一个共享变量的获取操作 S_A 和将对象释放允许别的处理机获取访问权的释放操作 S_R 。

12. 多指令流出主要受那些方面的限制？（2003）

- 1) 程序内在的指令级并行性。
- 2) 硬件实现的限制。
- 3) 超标量和超长指令字处理器固有的技术限制。

13. 简述 CISC 结构计算机的缺点和 RISC 结构计算机的设计原则。（2004）

CISC 结构的缺点：

- 1) 各种指令的使用频率相差悬殊。
- 2) 指令系统的复杂性带来了计算机体系结构的复杂性，增加了研制时间和成本。
- 3) 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) 许多复杂指令需要很复杂的操作，因而运行速度慢。
- 5) 由于各条指令的功能不均衡性，不利于采用先进的体系结构技术来提高系统性能。

RISC 结构计算机的设计原则：

- 1) 选取使用频率最高的那些指令，并补充一些最有用的指令。
- 2) 每条指令的功能应尽可能的简单，并在一个机器周期内完成。
- 3) 所有指令长度均相同。
- 4) 只用 load 和 store 操作指令访问存储器，其他指令操作均在寄存器之间进行。
- 5) 大多数指令都采用硬连逻辑来实现。
- 6) 以简单有效的方式支持高级语言。
- 7) 充分利用流水线技术来提高性能。

14. 给出在多处理机系统中进行时延隐藏的主要技术途径及其工作原理。（2001，2004，2007，2013）

MPP 系统采用了包括数据预取、相关性 Cache、松弛一致性、多现场这几种技术。

- 1) 数据预取：在数据使用之前就利用存储器的时机将其取到近处。
- 2) 相关性 Cache：减少对共享数据访问的竞争及时延。
- 3) 松弛一致性：在保证程序正确性前提下增加指令执行的并行。
- 4) 多现场：在产生时延时进行现场切换转而执行其他程序。

15. 简述“Cache-主存”层次和“主存-辅存”层次的区别。（2004）

比较项目	Cache-主存	主存-辅存
目的	为了弥补主存速度上的不足	为了弥补主存容量的不足
存储管理实现	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	几百比一
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
失效时 CPU 是否切换	不切换	切换到其他进程

16. 简述使用物理地址进行 DMA 所存在的问题，及其解决办法。（2004）

存在问题：

- 1) 对于超过一页的数据，由于缓冲区使用的页面在物理存储器中不一定是连续的，传输会发生问题。
- 2) DMA 正在存储器和帧缓冲器之间传输数据时，操作系统从存储器中移除一些页（或重新分配），DMA 将会在存储器中错误的页面上传输数据。

解决方法：

采用虚拟 DMA 技术，允许 DMA 设备直接使用虚拟地址，在 DMA 期间由硬件将虚拟地址映射到物理地址。I/O 使用的缓冲区在虚拟存储器中要求是连续的，物理页面可以分散在物理存储器中。如果使用虚拟 DMA 的进程在内存中被移动，操作系统能够及时地修改相应的 DMA 地址表，或者在 DMA 期间锁定存储器中相应的页面。

17. 栅栏（Barrier）同步怎样完成同步过程？（2011，2015）什么是栅栏（Barrier）同步？²⁰¹²在标准的栅栏（barrier）同步中，设单个处理器的通过时间（包括更新计数和释放锁）为 C，给出 N 个处理器一起进行一次同步所需要的时间表达式。（2004，2012，2014）在采用 k 元组合树的栅栏同步中，设单个处理器的通过时间（包括更新计数和释放锁）为 C，求 N 个处理器一起进行一次同步所需要的时间。（2005）

1) 栅栏同步过程

栅栏强制所有到达该栅栏的进程进行等待，直到全部的进程到达栅栏，然后释放全部的进程，从而实现同步。栅栏的典型实现是要用两个旋转锁，一个是用来记录到达栅栏的进程数，另一个是用来封锁栅栏，直到最后一个进程到达栅栏。栅栏的实现要不停地指定变量，直到它满足规定的条件。

2) 标准栅栏同步时间

忽略读写锁的时间。N 个处理器中的每一个都需要 C 个周期来锁住与栅栏相关的计数器并修改它的值，然后释放锁。考虑最坏的情况，所有 N 个处理器都要对计数器加锁并修改它的值。由于锁只能顺序访问计数器，在同一时间只能有一个处理器修改计数器的值，所以共需要花 NC 个时钟周期使得所有的处理器都到达栅栏。这是在默认存在排队锁提高栅栏的性能的情况下，若不存在排队锁，则 N 个处理器争用时，时间为 NC，N-1 个处理器争用时，时间为 (N-1)C，以此类推，则总的时间为 $NC + (N-1)C + (N-2)C + \dots + 2C + C$ ，即 $NC(N+1)/2$ 。

3) k 元组合树栅栏同步时间

k 元组合树是多个请求在局部结合起来形成树的一种分级结构，局部组合的分枝数量大大小于总分枝数量，因此组合树降低冲突的原因是将大冲突化解成为并行的多个小冲突。变量 k 表示扇入数目。树中每个结点组合 k 个进程，提供一个单独的计数器和锁，因而在每个结点有 k 个进程进行竞争，当第 k 个进程到达树中对应结点时则开始进行父结点，然后递增父结点的计数器，当父结点计数器到达 k 时，置 release 标志。每个结点计数器在最后一个进程到达时被初始化。所需时间为： $\lceil \log_k N \rceil * kC$ 。

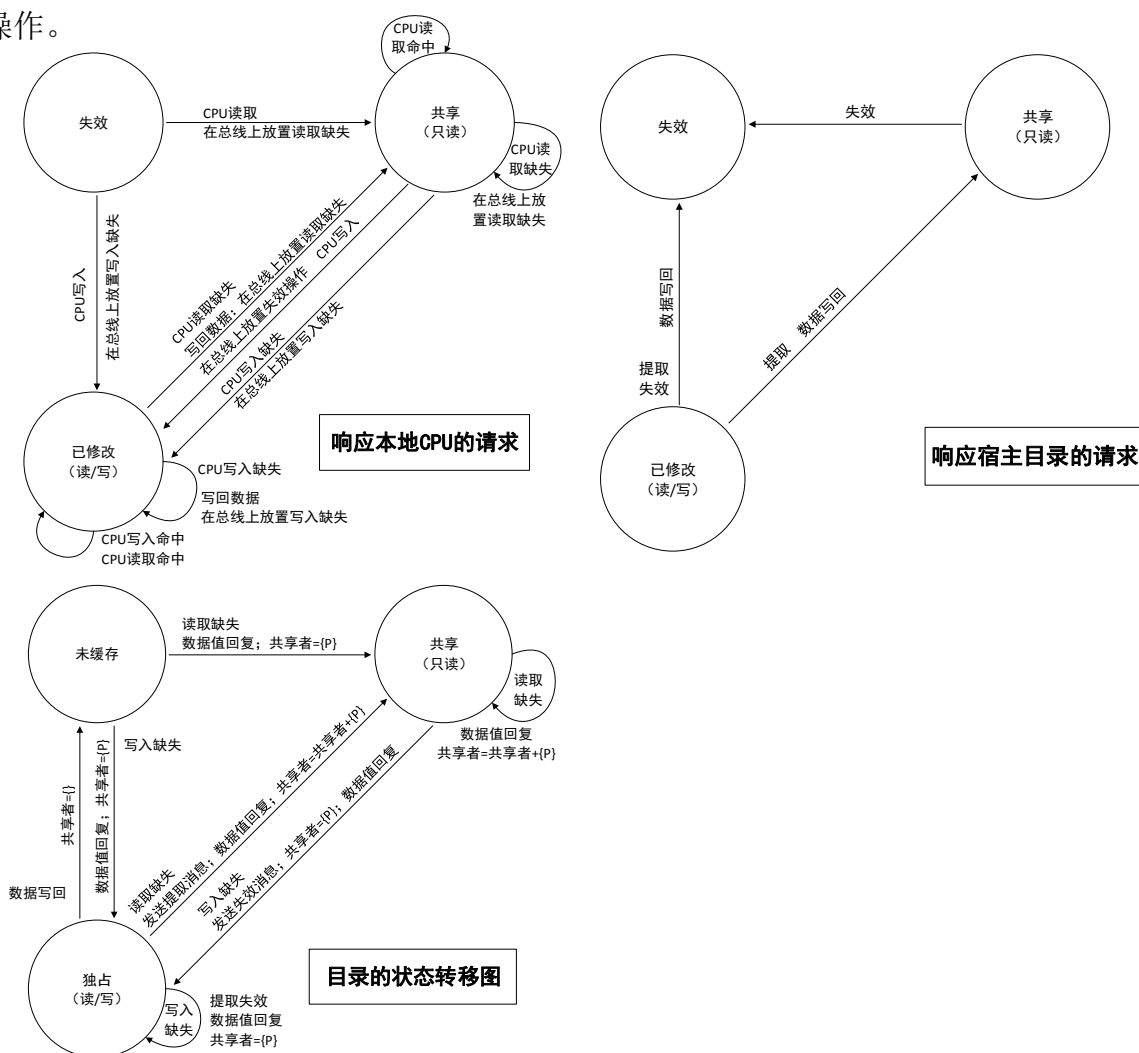
18. 在基于总线的集中共享多处理机系统中，Cache 相关性监听协议有三个状态：无效、共享

3) 写直达 Cache 与写回式 Cache 的主要区别

写直达 Cache 与写回式 Cache 最大的区别在于，本地处理器不需要读取另一个处理器脏 Cache 块。从而在写直达协议中不再提供硬件在读失效或写失效时将被替换的块的强制写回主存，也不再访问其他 Cache 拷贝而中断处理器访问。主存在 CPU 每次写 Cache 块时都会更新，所以在处理器产生读失效后就会直接访问主存，从主存中读取到正确的值。在写直达 Cache 中，有效的 Cache 块都与主存保持一致。

4) 目录协议的工作原理

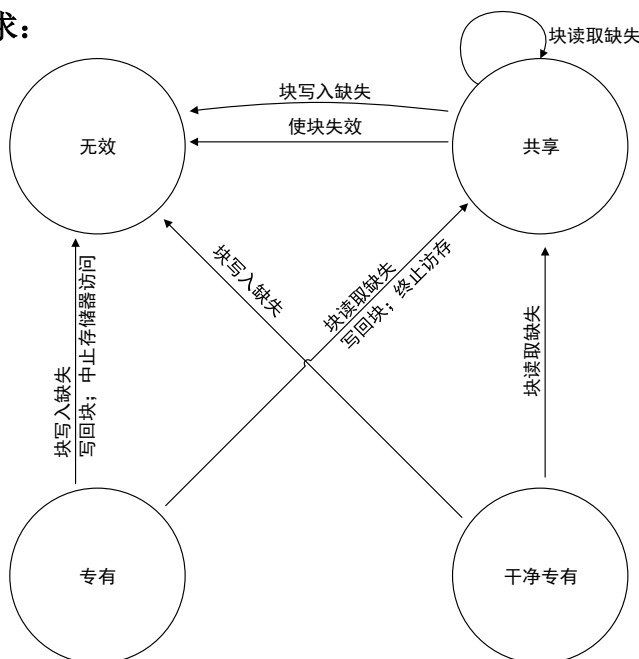
目录协议在每个结点增加了目录存储器用于存放目录。存储器的每一块在目录中对应有一项，每个目录项主要有“状态”和“位向量”两个内容。“状态”描述目录对应存储块的当前情况，“位向量”共有 N 位，每一位对应一个存储器的局部 Cache，用于指出该 Cache 中有无该块的副本。当处理器对某一块进行写操作时，根据位向量通知具有相应副本的处理器进行失效等操作。



19. 对同时多线程 (Simultaneous Multithreading)、粗粒度多线程和细粒度多线程比较，他们各有什么优缺点。(2006, 2010, 2013)

1) 粗粒度多线程：通过线程间的切换部分隐藏了代价较高、时延较长的阻塞带来的吞吐率的损失。

响应来自总线的请求:



21. 给出同时多线程（**Simultaneous Multithreading**）的并行工作原理及其在体系结构实现上的基础。（2008）如何在一个单线程的处理器上实现两路同时多线程？（2009）

1) 工作原理

同时多线程技术是一种在多流出、动态调度处理器上开发指令级并行和线程级并行的改进的多线程技术。同时多线程使用多发射和动态调度处理器同时实现指令级和线程级的并行，通过寄存器重命名和动态调度，来自各个独立线程多条指令可以被同时发射，而不考虑指令间的相关性，相关由动态调度负责处理。在同时多线程中，所有的发射槽在一个时钟周期内被多个线程共享，线程级并行和指令级并行被同时开发。

2) 结构基础

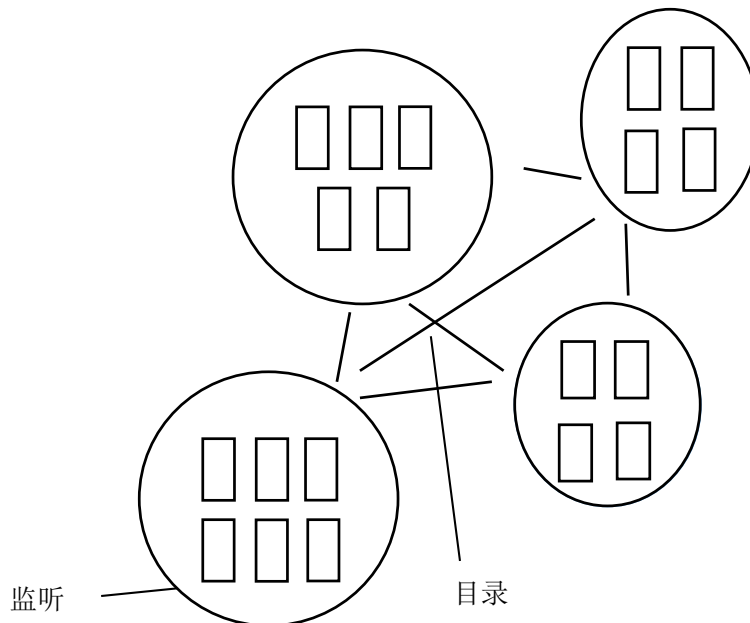
使用动态调度技术的处理器已经具有了开发线程级并行所需的硬件基础。具体来说，动态调度超标量处理器有大量的虚拟寄存器组，可以用来保存每个独立线程的寄存器状态。由于寄存器重命名机制提供了唯一的寄存器标识符，多个线程的指令可以在数据路径上混合执行，而不会导致各线程间源操作数和目的操作数的混乱。

通过乱序执行机制，可以很好地利用硬件功能，提高并行性。利用重排序缓存，将来自独立线程的指令以独立的方式提交。在以上描述的硬件基础上，我们只需通过在支持单线程的处理器上，为每个线程的实现设置重命名表，保留各自的 PC 值，提供多个线程的指令结果提交的能力来实现。

3) 具体实现

- ①设置多个独立的重命名表，以保存每个独立线程的寄存器状态；
- ②让取值部件和指令 Cache 并发预取多个线程指令流，并分别设置各自的 PC；
- ③指令完成时，处理器为不同线程提供指令结果提交。

22. 一台超结点的分布共享多处理机，其 Cache 相关性（coherency）协议是结点内部运行监听协议，结点之间运行目录协议。试述系统解决相关性的工作原理。（2008，2009，2013）



超结点的分布共享多处理机在结点内部运行监听协议，结点之间运行目录协议。

即在超结点内部每个处理机的Cache除了包含物理存储器中的块的数据副本之外，也保存着各个块的共享状态信息。Cache通常连在共享存储器的总线上，各个Cache控制器通过监听总线来判断它们是否有总线请求的数据块。

每个超结点中使用一个目录存储器用于存放目录，存储器的每一块在目录对应有一项，每一个目录项主要有“状态”和“位向量”两种成分。“状态”描述该目录所对应存储块的当前情况；“位向量”共有N位，其每一位对应于一个超结点，用于指出该超结点中是否有存储块的拷贝。

当处理器进行写时，必须先获得总线的控制权，然后将要作废的数据块地址放在总线上。超结点内其他处理机一直监听总线，检测该地址是否存在于它们的Cache中。若存在，则作废相应的数据块。同时，向该块所在的目录发消息，目录根据向量位判断该块是否存在于其它的超结点中，若存在，则根据位向量通知所有具有相应拷贝的超结点进行作废操作。超结点在接收到目录发来的写作废消息后，将该要作废的数据块地址放在总线上。当处理器写未命中时，除了将其它处理器对应的Cache块作废外，还需要取回该数据块。

监听过程的实现可利用Cache中块的标志位。每个块的有效位使作废机制的实现较为容易。由写作废或其他事件引起的失效处理很简单，只需要将该位设置成无效即可。而对于写，我们希望知道时候别的Cache中是否也有此数据的共享拷贝。因为如果没有别的Cache拷贝，则无需将写地址放在写回Cache的总线上，这样可降低所用的时间和所需的带宽。为了区别某个数据块是否为共享，需要给每个Cache块加上一个特殊的状态位来说明它是否为共享。当对共享块进行写时，本Cache将写作废的请求放在总线上，Cache块状态由共享变为非共享或专有，同时目录向其它超结点发送写作废消息，目录中该块的状态变为专有。如果另一个处理器要求访问此块，则该块在Cache标志位状态会再转化为共享，若是其它超结点中的处理器访问，则目录中的状态变为共享。目录必须跟踪每个数据块在超结点中的状态，这些状态包括共享、未缓冲和专有。

23. Tell me the advantages and disadvantages of hardware-based speculation (前瞻)。(2010,

2015)

前瞻技术实现的关键是允许指令的乱序执行，但必须循序确认，它不仅能够得到更高的并行性，还能在处理器未判断指令是否可以执行前就提前执行，以克服控制相关。基于硬件的前瞻和动态调度相结合，可以做到同一种体系结构，实现不同机器能够使用相同的编译器。

具体优点有：1) 动态识别访存地址；2) 具有较高的推断准确性；3) 精确断点模型，可以进行精确异常处理；4) 无需附加代码；5) 有良好的二进制特性，无需重新编译。

主要缺点是支持前瞻的硬件太复杂，需要大量硬件资源。

24. 大规模机器的同步有哪些软件和硬件支持方法？（2011）

软件方法：

- 1) 旋转锁：是指处理器环绕一个锁不停地旋转而请求获得该锁，当锁的占用时间很少以及加锁过程延迟很低时可以采用旋转锁。
- 2) 软件排队锁：可以排队记录等待的进程，当锁释放时发送出一个已确定的等待进程。软件实现主要是采用记录等待进程的排队数组进行排队。
- 3) 组合树：是多个请求在局部结合起来形成树的一种分级结构，局部组合的分枝数量大大小于总的分枝数量，因此组合树降低冲突的原因是将大冲突化解成为并行的多个小冲突。

硬件方法：

- 1) 硬件排队锁：可以排队记录等待的进程，当锁释放时发送出一个已确定的等待进程。硬件实现一般是在基于目录的机器上，通过硬件向量等方式来进行排队和同步控制。
- 2) 硬件原语：用硬件实现，能够以原子方式自动读出和修改存储单元。