

图8-2 三类基于总线的多处理机：a) 没有高速缓存；b) 有高速缓存；c) 有高速缓存与私有存储器

高速缓存上得到满足，总线流量就大大减少了，这样系统就能够支持更多的CPU。一般而言，高速缓存不以单个字为基础，而是以32字节或64字节块为基础。当引用一个字时，它所在的整个数据块（叫做一个cache行）被取到使用它的CPU的高速缓存当中。

每一个高速缓存块或者被标记为只读（在这种情况下，它可以同时存在于多个高速缓存中），或者标记为读写（在这种情况下，它不能在其他高速缓存中存在）。如果CPU试图在一个或多个远程高速缓存中写入一个字，总线硬件检测到写，并把一个信号放到总线上通知所有其他的高速缓存。如果其他高速缓存有个“干净”的副本，也就是同存储器内容完全一样的副本，那么它们可以丢弃该副本并让写者在修改之前从存储器取出高速缓存块。如果某些其他高速缓存有“脏”（被修改过）副本，它必须在处理写之前把数据写回存储器或者把它通过总线直接传送到写者上。高速缓存这一套规则被称为高速缓存一致性协议，它是诸多协议之一。

还有另一种可能性就是图8-2c中的设计，在这种设计中每个CPU不止有一个高速缓存，还有一个本地的私有存储器，它通过一条专门的（私有）总线访问。为了优化使用这一配置，编译器应该把所有程序的代码、字符串、常量以及其他只读数据、栈和局部变量放进私有存储器中。而共享存储器只用于可写的共享变量。在多数情况下，这种仔细的放置会极大地减少总线流量，但是这样做需要编译器的积极配合。

## 2. 使用交叉开关的UMA 多处理机

即使有最好的高速缓存，单个总线的使用还是把UMA多处理机的数量限制在16至32个CPU。要超过这个数量，需要不同类型的互连网络。连接 $n$ 个CPU到 $k$ 个存储器的最简单的电路是交叉开关，参见图8-3。交叉开关在电话交换系统中已经采用了几十年，用于把一组进线以任意方式连接到一组出线上。

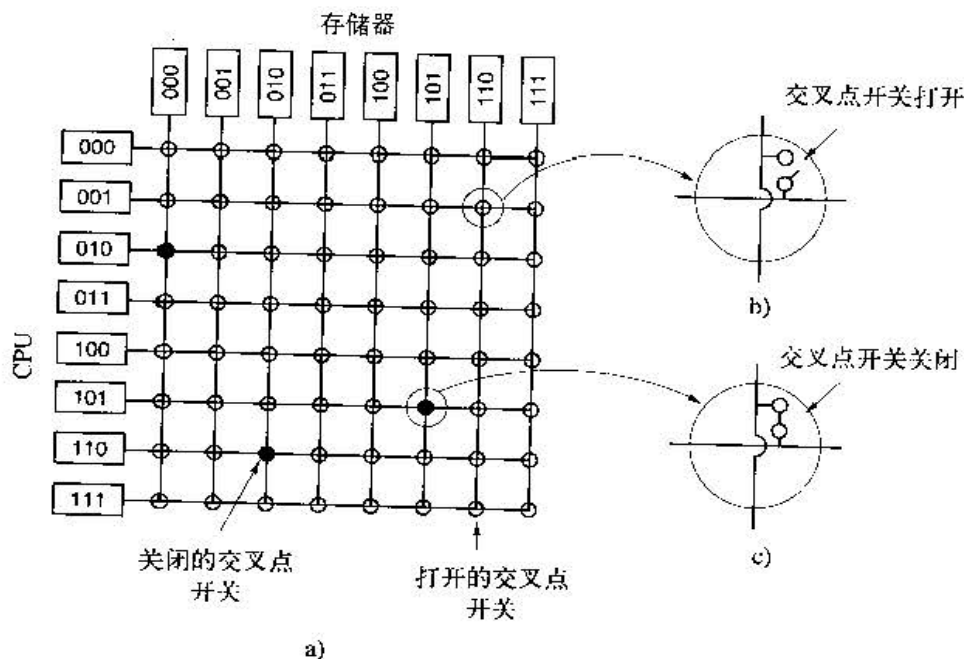


图8-3 a) 8x8交叉开关；b) 打开的交叉点；c) 闭合的交叉点