

程序的程序员来说,我们建议在心里采用一个使用写回和写分配的高速缓存的模型。这样建议有几个原因。通常,由于较长的传送时间,存储器层次结构中较低层的缓存更可能使用写回,而不是直写。例如,虚拟内存系统(用主存作为存储在磁盘上的块的缓存)只使用写回。但是由于逻辑电路密度的提高,写回的高复杂性也越来越不成为阻碍了,我们在现代系统的所有层次上都能看到写回缓存。所以这种假设符合当前的趋势。假设使用写回写分配方法的另一个原因是,它与处理读的方式相对称,因为写回写分配试图利用局部性。因此,我们可以在高层次上开发我们的程序,展示良好的空间和时间局部性,而不是试图为某一个存储器系统进行优化。

6.4.6 一个真实的高速缓存层次结构的解剖

到目前为止,我们一直假设高速缓存只保存程序数据。不过,实际上,高速缓存既保存数据,也保存指令。只保存指令的高速缓存称为 *i-cache*。只保存程序数据的高速缓存称为 *d-cache*。既保存指令又包括数据的高速缓存称为统一的高速缓存(unified cache)。现代处理器包括独立的 *i-cache* 和 *d-cache*。这样做有很多原因。有两个独立的高速缓存,处理器能够同时读一个指令字和一个数据字。*i-cache* 通常是只读的,因此比较简单。通常会针对不同的访问模式来优化这两个高速缓存,它们可以有不同的块大小,相联度和容量。使用不同的高速缓存也确保了数据访问不会与指令访问形成冲突不命中,反过来也是一样,代价就是可能会引起容量不命中增加。

图 6-38 给出了 Intel Core i7 处理器的高速缓存层次结构。每个 CPU 芯片有四个核。每个核有自己私有的 L1 *i-cache*、L1 *d-cache* 和 L2 统一的高速缓存。所有的核共享片上 L3 统一的高速缓存。这个层次结构的一个有趣的特性是所有的 SRAM 高速缓存存储器都在 CPU 芯片上。

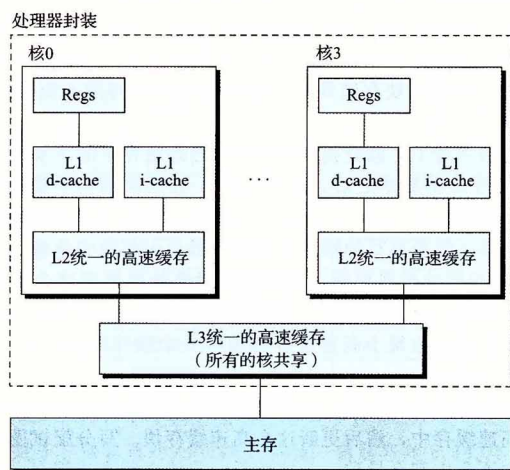


图 6-38 Intel Core i7 的高速缓存层次结构

图 6-39 总结了 Core i7 高速缓存的基本特性。