**主存到Cache直接映射、全相联映射和组相联映射**

---- Cache的容量很小，它保存的内容只是主存内容的一个子集，且Cache与主存的数据交换是以块为单位的。

---- 为了把信息放到Cache中，必须应用某种函数把主存地址定位到Cache中，这称为**地址映射**。

---- 在信息按这种映射关系装入Cache后，CPU执行程序时，会将程序中的主存地址变换成Cache地址，这个变换过程叫做地址变换。

Cache的地址映射方式有直接映射、全相联映射和组相联映射。

假设某台计算机**主存容量**为1MB，被分为**2048**块，每块512B；Cache容量为8KB，被分为**16**块，每块也是512B。下面以此为例介绍三种基本的地址映射方法。

1. 直接映射

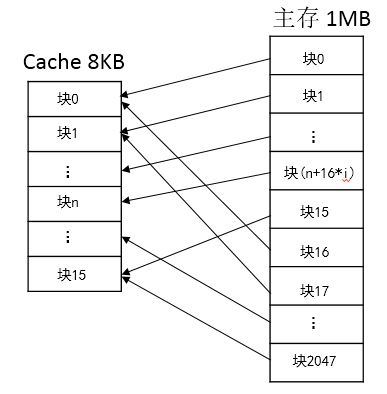
---- 直接映射的Cache组织如图（1）所示。主存中的一个块只能映射到Cache的**某一特定块**中去。例如，

主存的第0块、第16块、第32块、第48块、……、第2032块等128块，只能映射到Cache的第0块；

主存的第1块、第17块、第33块、第49块、……、第2033块等128块，只能映射到Cache的第1块；

以此类推，主存的第15块、第31块、第47块、……、第2047块等128块，只能映射到Cache的第15块中。

映射完毕，Cache总共有0～15即16块，主存中的每128（2048/16）块，只能映射到Cache中的某一个块中。

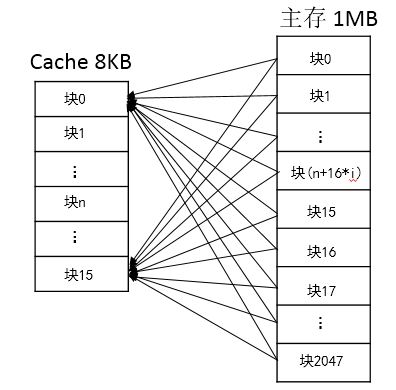


                                          图（1）

直接映射是最简单的地址映射方式，它的硬件简单，成本低，地址变换速度快，而且不涉及替换算法问题。但是这种方式不够灵活，Cache的存储空间得不到充分利用，每个主存块只有一个固定位置可存放，容易产生冲突，使Cache效率下降，因此只适合大容量Cache采用。例如，如果一个程序需要重复引用主存中第0块与第16块，最好将主存第0块与第16块同时复制到Cache中，但由于它们都只能复制到Cache的第0块中去，即使Cache中别的存储空间空着也不能占用，因此这两个块会不断地交替装入Cache中，导致命中率降低。

2. 全相联映射

---- 图（2）是全相联映射的Cache组织，主存中任何一块都可以映射到Cache中的任何一块位置上。



                                         图（2）

全相联映射方式比较灵活，主存的各块可以映射到Cache的任一块中，Cache的利用率高，块冲突概率低，只要淘汰Cache中的某一块，即可调入主存的任一块。

但是，由于Cache比较电路的设计和实现比较困难，这种方式只适合于**小容量**Cache采用。

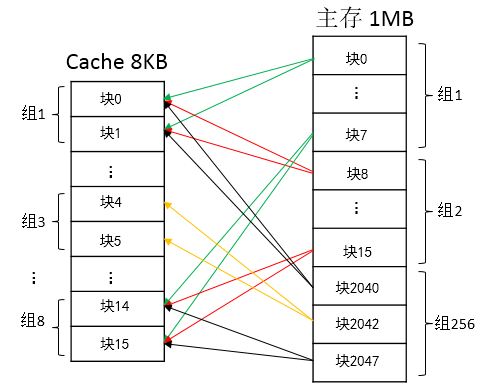
3. 组相联映射

---- 组相联映射实际上是直接映射和全相联映射的折中方案，其组织结构如图（3）所示。

主存和Cache都**分组**，主存中一个**组内的块数**与Cache中的**分组数**相同，组间采用直接映射，组内采用全相联映射。

也就是说，将Cache分成u组，每组v块，主存块存放到哪个组是固定的，至于存到该组哪一块则是灵活的。

例如，Cache分为8组，每组2块，主存分为256组，每组8块。



                                                  图（3）

主存中的各块与Cache的组号之间有固定的映射关系，但可自由映射到对应Cache组中的任何一块。例如：

主存的第0块、第8块、第16块、…第2040块等**256**块均映射于Cache的第1组，但可映射到其中的**第0块或第1块**；

主存的第1块、第9块、第17块、…第2041块等均映射于Cache的第2组，但可映射到Cache第2组中的第2块或第3块；

主存的第2块、第10块、第18块、…第2042块等均映射于Cache的第3组，但可映射到Cache第3组中的第4块或第5块；

主存的第7块、第15块、第23块、…第2047块等均映射于Cache的第8组，但可映射到Cache第8组中的第14块或第15块。

常采用的组相联结构Cache，每组内有2、4、8、16块，称为2路、4路、8路、16路组相联Cache。以上为2路组相联Cache。

组相联结构Cache是前两种方法的折中方案，适度兼顾二者的优点，尽量避免二者的缺点，因而得到普遍采用。

**实例分析：**

1.容量为64块的Cache采用组相联方式映像，字块大小为128字节，每4块为一组，若主容量为4096块，且以字编址，那么主存地址为（19）位，主存区号为（6）位。

**解**：

组相联的地址构成为：区号+组号+块号+块内地址。

主存的**每个分区/组大小**与整个Cache大小相等，故此主存需要分的区数为：4096/64=64，因为26＝64，因此需要6位来表示区号。（直接相联？）

每4块为一组，故共有组数 64/4 = 16 ，因为24＝16，因此需要4位表示组号。每组4块，故表示块号需要2位。

块内地址共128字节，27＝128，所以块内地需要7位表示。所以：主存地址的位数＝6+4+2+7 ＝ 19

主存区号的位数＝6

**个人见解**：Cache有S组，每组有E块，即S = 16，E = 4，Cache大小：64块×128B = **8KB**。

主存大小：4096×128B = 2^12\*2^7 = 2^19,即主存地址有**19位**。4096/64= 2^6主存区号为6位。

2.某 32 位计算机的 cache 容量为 16KB，cache 块的大小为 16B，若主存与 cache 的地址映射采用直接映射方式，则主存地址为 1234E8F8（十六进制）的单元装入的 cache 地址为 **C** 。

A. 00 0100 0100 1101 (二进制)  
B. 01 0010 0011 0100 (二进制)  
C. 10 1000 1111 1000 (二进制)  
D. 11 0100 1110 1000 (二进制)

**解**：Cache大小为16KB，块大小为16B，所以Cache被分成16KB/16B=1024块，因210＝1024故需要**10位来表示块数**。

24＝16故块内地址需要4位来表示。所以Cache的地址线位置为14位。

由于采用直接映像的方式，所以主存的后14位就是要装入的到Cache中的位置。故选 C.

个人见解：Cache的容量是16KB = 16×1024B = 16384B，主存的地址为0x1234E8F8/（16×1024B）= 18643. 该地址对应的是主存的第18643块。

根据公式：cache的块地址i = 主存的块地址 % 16384 = 1234E8F8 - 1234C000 = 0x28F8 选C

# Cache与主存之间的全相联映射，直接映射和组相联映射的区别

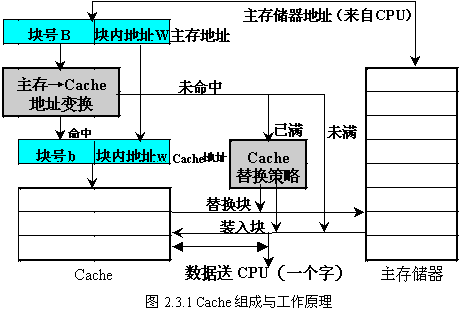
转载 2015年04月06日 11:23:30

Cache与主存之间的全相联映射，直接映射和组相联映射的区别

1.高速缓冲存储器的功能、结构与工作原理

　　高速缓冲存储器是存在于主存与CPU之间的一级存储器， 由静态存储芯片(SRAM)组成，容量比较小但速度比主存高得多，接近于CPU的速度。 Cache的功能是用来存放那些近期需要运行的指令与数据。目的是提高CPU对存储器的访问速度。为此需要解决2个技术问题：一是主存地址与缓存地址的映象及转换； 二是按一定原则对Cache的内容进行替换。

Cache的结构和工作原理如图2.3.1所示。



主要由三大部分组成：

Cache存储体：存放由主存调入的指令与数据块。

　　地址转换部件：建立目录表以实现主存地址到缓存地址的转换。

　　替换部件：在缓存已满 时按一定策略进行数据块替换，并修改地址转换部件。

2.地址映象与转换

　　地址映象是指某一数据在内存中的地址与在缓冲中的地址，两者之间的对应关系。下面介绍三种地址映象的方式。

1.全相联方式

　　地址映象规则：主存的任意一块可以映象到Cache中的任意一块

(1) 主存与缓存分成相同大小的数据块。

(2) 主存的某一数据块可以装入缓存的任意一块空间中。

　　全相联方式的对应关系如图2.3.2所示。如果Cache的块数为Cb，主存的块数为Mb，则映象关系共有Cb×Mb种。

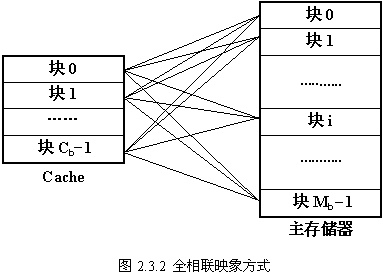
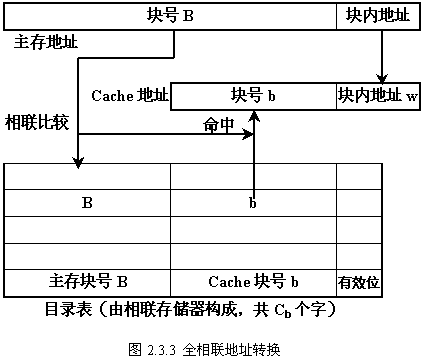
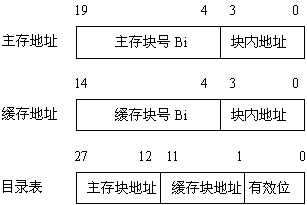


图2.3.3示出了目录表的格式及地址变换规则。 目录表存放在相关（联）存储器中，其中包括三部分：数据块在主存的块地址、存入缓存后的块地址、及有效位（也称装入位）。由于是全相联方式，因此，目录表的容量应当与缓存的块数相同。



举例：某机主存容量为1M，Cache的容量为32KB， 每块的大小为16个字（或字节）。 划出主、缓存的地址格式、 目录表格式及其容量。



　　容量：与缓冲块数量相同即211＝2048（或32K/16＝2048）。

优点：命中率比较高，Cache存储空间利用率高。

缺点：访问相关存储器时，每次都要与全部内容比较，速度低，成本高，因而应用少。

2.直接相联方式

　　地址映象规则： 主存储器中一块只能映象到Cache的一个特定的块中。

(1) 主存与缓存分成相同大小的数据块。

(2) 主存容量应是缓存容量的整数倍，将主存空间按缓存的容量分成区，主存中每一区的块数与缓存的总块数相等。

(3) 主存中某区的一块存入缓存时只能存入缓存中块号相同的位置。

　　图2.3.4示出了直接相联映象规则。 可见，主存中各区内相同块号的数据块都可以分别调入缓存中块号相同的地址中，但同时只能有一个区的块存入缓存。由于主、缓存块号相同，因此，目录登记时，只记录调入块的区号即可。

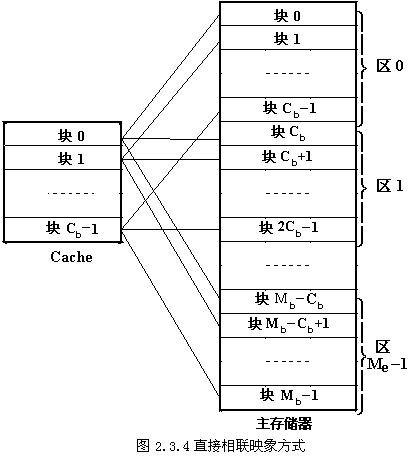
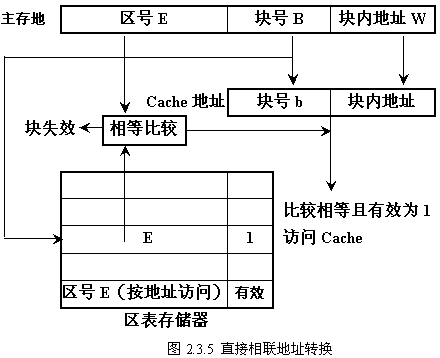


　　图2.3.5示出了主、 缓冲地址格式、目录表的格式及地址变换规则。主、缓存块号及块内地址两个字段完全相同。目录表存放在高速小容量存储器中，其中包括二部分：数据块在主存的区号和有效位。目录表的容量与缓存的块数相同。



地址变换过程：用主存地址中的块号B去访问目录存储器， 把读出来的区号与主存地址中的区号E进行比较， 比较结果相等，有效位为1，则Cache命中，可以直接用块号及块内地址组成的缓冲地址到缓存中取数；比较结果不相等，有效位为1， 可以进行替换，如果有效位为0，可以直接调入所需块。

优点：地址映象方式简单，数据访问时，只需检查区号是否相等即可，因而可以得到比较快的访问速度，硬件设备简单。

缺点：替换操作频繁，命中率比较低。

　　举例：上例中，主存容量为1M， Cache的容量为32KB，每块的大小为16个字（或字节）。划出主、缓存的地址格式、目录表格式及其容量。

　　容量：与缓冲块数量相同即211＝2048（或32K/16＝2048）。

3.组相联映象方式

　　组相联的映象规则：

(1) 主存和Cache按同样大小划分成块。

(2) 主存和Cache按同样大小划分成组。

(3) 主存容量是缓存容量的整数倍，将主存空间按缓冲区的大小分成区，主存中每一区的组数与缓存的组数相同。

(4) 当主存的数据调入缓存时，主存与缓存的组号应相等，也就是各区中的某一块只能存入缓存的同组号的空间内，但组内各块地址之间则可以任意存放， 即从主存的组到Cache的组之间采用直接映象方式；在两个对应的组内部采用全相联映象方式。

　　图2.3.6示出了组相联的映象关系， 图中缓存共分Cg个组，每组包含有Gb块； 主存是缓存的Me倍，所以共分有Me个区， 每个区有Cg组，每组有Gb块。那么， 主存地址格式中应包含4个字段：区号、区内组号、组内块号和块内地址。 而缓存中包含3个字段：组号、组内块号、块内地址。主存地址与缓存地址的转换有两部分，组地址是按直接映象方式，按地址进行访问，而块地址是采用全相联方式，按内容访问。组相联的地址转换部件也是采用相关存储器实现，见图2.3.7。

　　相关存储器中每个单元包含有： 主存地址中的区号E与组内块号B，两者结合在一起，其对应的字段是缓存块地址b。相关存储器的容量，应与缓存的块数相同。当进行数据访问时，先根据组号，在目录表中找到该组所包含的各块的目录，然后将被访数据的主存区号与组内块号，与本组内各块的目录同时进行比较。如果比较相等，而且有效位为“1”则命中。

  将其对应的缓存块地址b送到缓存地址寄存器的块地址字段，与组号及块内地址组装即形成缓存地址。如果比较不相等，说明没命中，所访问的数据块尚没有进入缓存，则进行组内替换；如果有效位为0，则说明缓存的该块尚未利用， 或是原来数据作废，可重新调入新块。

　　优点：块的冲突概率比较低，块的利用率大幅度提高，块失效率明显降低。

　　缺点：实现难度和造价要比直接映象方式高。