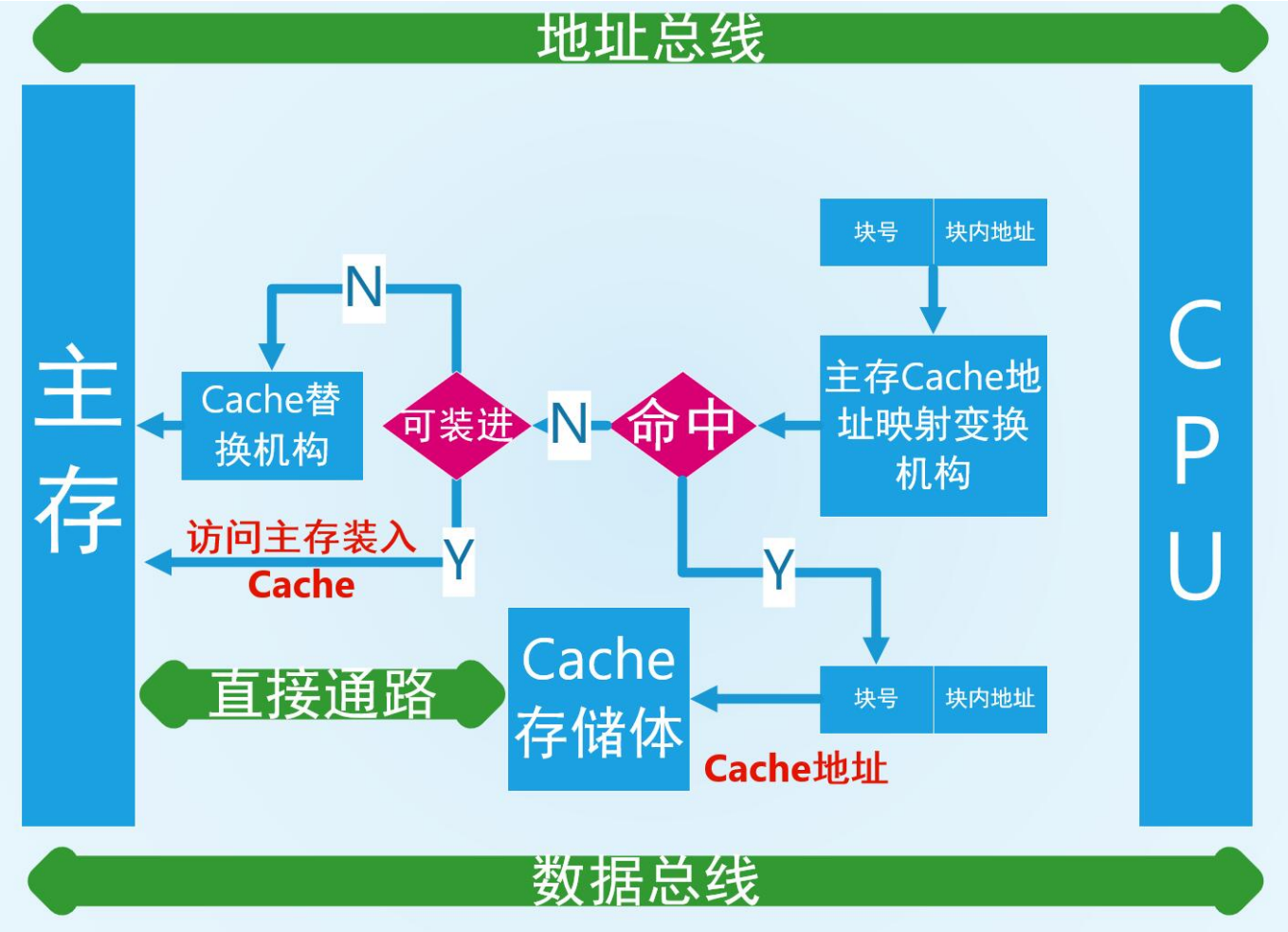
高速缓冲器

Cache的出现是为了解决CPU和主存速度不匹配的问题；

Cache位于存储器层次结构的顶层，通常由SRAM组成；

工作原理



统一缓存：指令和数据都放在统一缓存中

分立缓存：指令和数据分别存放在两个缓存中 数据Cache、指令Cache

Cache中主存块的替换算法

- 随机算法
- 先进先出算法
- 近期最少使用
- 最不经常使用

近期最少使用

LRU算法对每行设置一个计数器，Cache每命中一次，命中行计数器清0，而其他各行计数器均加1；

需要替换时比较各特定行的计数值，将计数值最大的换出；

最不经常使用

新建行后从0开始计数，每访问一次，被访问行的计数器加1；

需要时比较各特定行的计数值，将计数值最小的行换出

写策略

Cache写命中

- 全写法；当CPU对Cache写命中时，必须把数据同时写入Cache和主存；当某一块需要替换时，不必把该块写入主存，直接替换即可；
- 写回法；当CPU对Cache写命中时，只修改Cache的内容，而不立即写入内存，只有当此块被换出时才写回主存；每个Cache行需要设置一个脏（标志）位，以标志此块是否被CPU修改过；

Cache写不命中

- 写分配法：加载主存中的块到Cache中，然后更新这个Cache块
- 非写分配法：只写入内存，不进行调块

写分配法通常和写回法合用

非写分配法通常与全写法合用

Cache和主存的映射方式

由主存地址映射到Cache地址称为地址映射；

即把存放在主存中的数据按照某种规则装入Cache

由于Cache的块数比主存块少，因此需要加入主存字块标记和字块内地址

有直接映射、全相联映射、组相联映射

直接映射（对号入座）

- 主存字块标记：指明当前Cache是主存中哪一块的副本
- Cache字块地址：指明存放在Cache中的哪个Cache块
- 字块内地址：该字在Cache块中的地址

直接映射关系：j = i mod 2^c

j是Cache的块号，i是主存的块号，2^c是Cache的总块数

主存数据块只能装入Cache中唯一位置；

若这位置已有内容，则发生块冲突，原先的块将被无条件替换出去（无需使用替换算法）

全相联映射（随便坐）

- 主存字块标记：指明当前Cache是主存中哪一块的副本
- 字块内地址：由于Cache块和主存块都是由若干字节组成的，当需要访问某个字时，需要找到该字在指定的Cache块中的地址

可以把主存数据装入Cache中任何位置；

优点

- 比较灵活
- Cache块的冲突概率低
- 空间利用率高
- 命中率高

缺点

- 地址变换速度慢
- 实现成本高

组相联映射（按号分组，组内随便坐）

将Cache空间分成大小相同的组，主存中的一个数据块可以装入一组内任何一个位置；

组间采用直接映射，主存数据块只能装入Cache中唯一的组；

组内采用全相联映射，主存数据块可以装入组内的任意位置；

- 把Cache分成Q组，每组有R块
- 组数 = Cache总容量 / 组容量
- 组容量 = R（块数） * 块容量
- i = j mod Q 属于哪一组