

存储系统

存储系统的层次结构

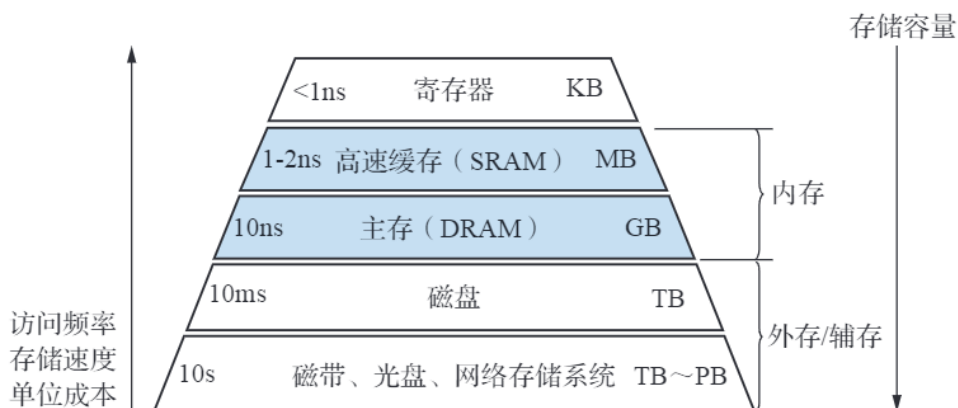


图 4.1 存储系统的分级结构

- 存储容量（能存储多少二进制位）表示法有位表示法和字节表示法
 - $256K \times 8$ 位表示该芯片有 $256K$ 个单元, 每个存储单元的长度为8个二进制位
 - $128B$ 表示该芯片有128个单元
- 存储周期与存储时间
- $SRAM$ n 根地址线, 存储容量为 2^n 个存储单元; $DRAM$ n 根地址线, 存储容量为 2^{2n} 个存储单元

主存中数据的存放

- 两个字长
 - 存储字长（一个存储字的字长，一般对于计算机来说是固定的）：主存一个存储单元所存储的二进制位数, 一般是字节的整倍数(如16位, 32位, 64位)
 - 机器字长(简称字长，华科教材称为数据字长)：计算机一次能处理的二进制数的位数
 - 两个字长不一定相同, 机器字长为32位的计算机采用的存储字长可以是16位, 32位和64位
- 大端和小端存放方式。主流处理器都采用小端存放方式，即低字节地址单元存放数据的低字节
- 边界对齐（数据起始字节地址末位几位为0的问题）
- 存储器扩展（看历年计组卷的真题，尤其是22和23的）
- 主存一般是按字节编址
- 字、半字、双字、字节寻址（字就是WORD，简称W）
- 存储容量与地址位数的关系，假设有16MB的内存，则需要24位地址，因为 $2^{24} = 16MB$
- CPU每次访存都是取出一个存储字

高速缓冲存储器 (Cache)

工作流程

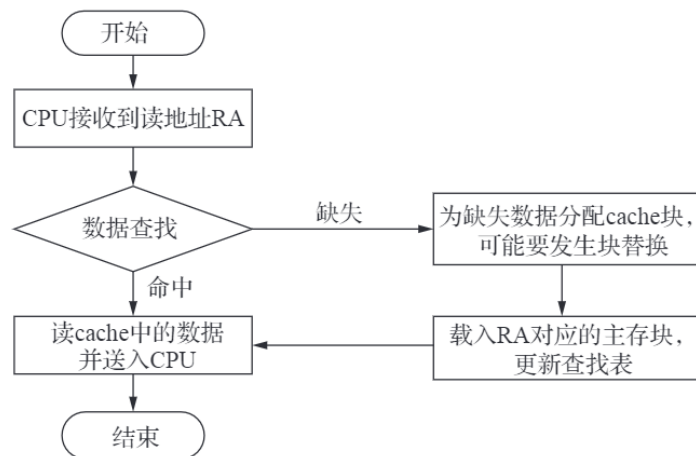


图 4.33 cache 读操作的基本流程

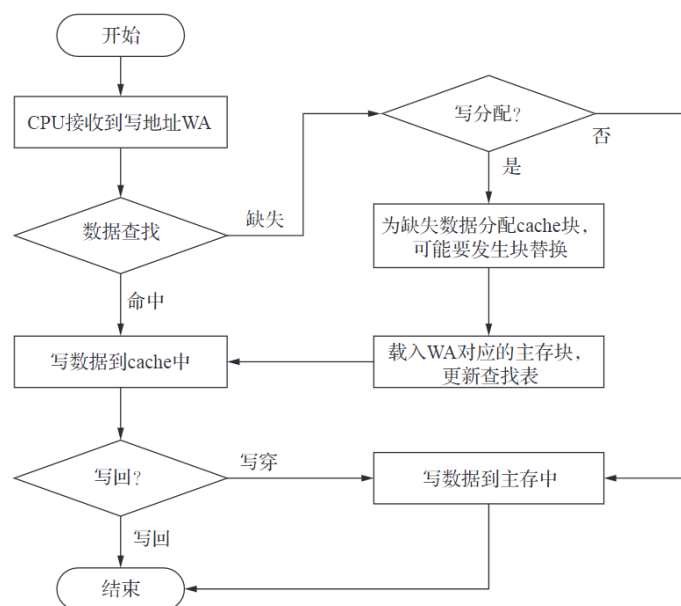


图 4.34 cache 写操作的基本流程

基本概念

- 增加Cache后，CPU访问数据时不是直接就访问主存，而是先通过字节地址访问Cache，通过查找机制判断要访问的数据是否在Cache中，如果在，则称为**数据命中 (Hit)**，如果不在，则称为**数据缺失 (Miss)**，需要将缺失数据从主存调入Cache中才能访问数据
- 学会计算命中率：
$$\frac{\text{命中Cache次数}}{\text{命中Cache次数} + \text{访问主存信息次数}}$$
- Cache中数据块大小与主存中每个数据块大小是一样的

地址映射机制

重点是学会计算行索引、组索引以及 *offset* 的位数。一般来说，按照字节编址，假设块大小为 n B，则 *offset* 位数为 $\log_2 n$ 。

行索引以及组索引的计算方式在下面。

还要学会利用主存地址计算块号和行号（通常是利用数组）

提高Cache命中率的方法：增大Cache数据块的容量、提高程序访问的局部性

- 全相联
 - 主存中的每一个块都可以放置到Cache中任意一行中的数据块中
 - Cache每个数据块大小为 2^w 字节, Cache的实际容量为 $n \times (1 + s + 8 \times 2^w)$ 位

- 特点: 利用率高

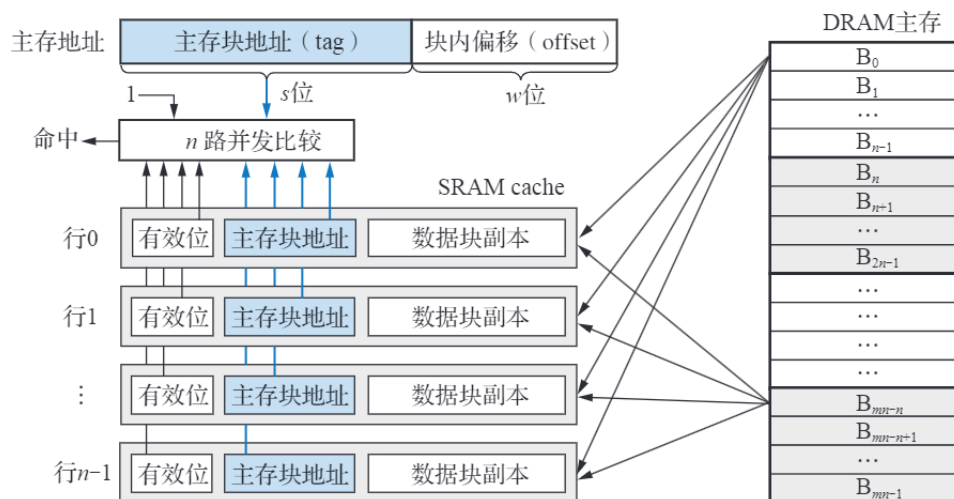


图 4.36 全相联映射逻辑示意图

直接相联

- 每一个主存数据块只能映射到Cache中固定的行, 映射规则为:
 $Cache \text{ 行号} = \text{主存块号} \bmod (Cache \text{ 行数 } n)$
- 主存中每个分区包含的块数与Cache行数相等, 每个分区中每个块映射的行不相同
- Cache数据块大小为 2^w 字节, Cache行数 $n = 2^r$, Cache实际容量为 $n \times (1 + s - r + 8 \times 2^w)$
- 利用率低(命中率低) 冲突率高 不需要替换算法(直接替换)

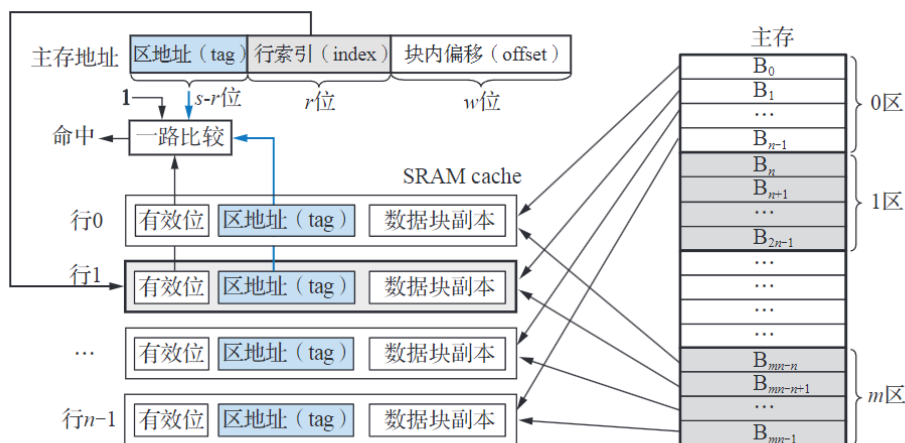


图 4.38 直接相联映射逻辑示意图

组相联

- 每一个主存数据块只能映射到Cache中固定的组(可以是任一行), 映射规则为:
 $Cache \text{ 组号} = \text{主存块号} \bmod (Cache \text{ 组数 } n)$
- 以图中二路组相联为例, Cache数据块大小为 2^w 字节, Cache组数 $n = 2^d$, Cache实际容量为 $2n \times (1 + s - d + 8 \times 2^w)$

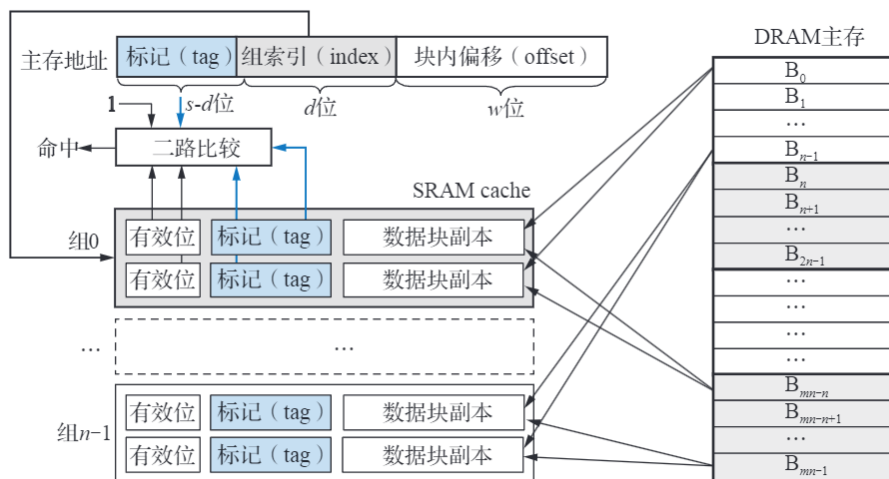


图 4.41 组相联映射逻辑示意图

虚拟存储器（页式虚拟存储器）

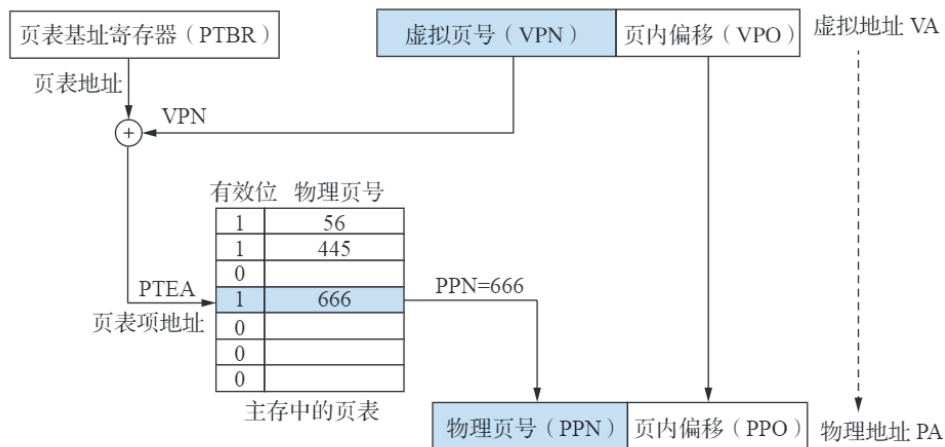


图 4.48 基于页表的虚拟地址与物理地址转换过程

- 物理页和虚拟页大小相同，因此 VPO 与 PPO 位数相同。 VPN 与 PPN 的大小取决于虚拟空间和主存空间的容量
- PTE ：页表项，包括有效位和 PPN 。页表项的数目为： 2^{VPN}
- $PTEA$ ：页表项地址
- 页表放在主存/Cache中
- 页表项的有效位为1，则当前页的数据在主存中；否则可能是虚拟页未分配页或缺页异常（后者需要将磁盘上的页载入主存，同时更新页表项）

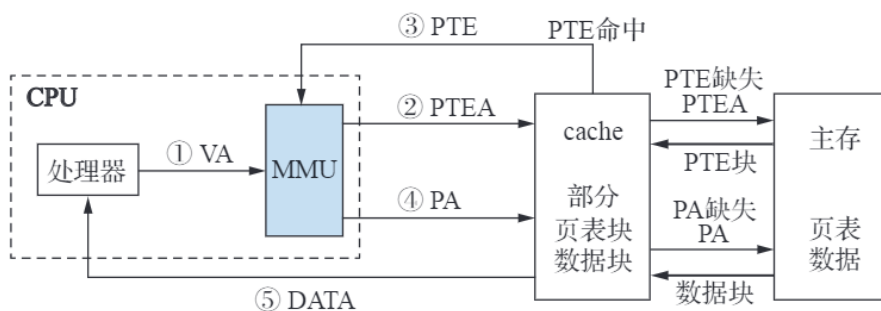


图 4.50 结合 cache 的页式虚拟存储器访问流程

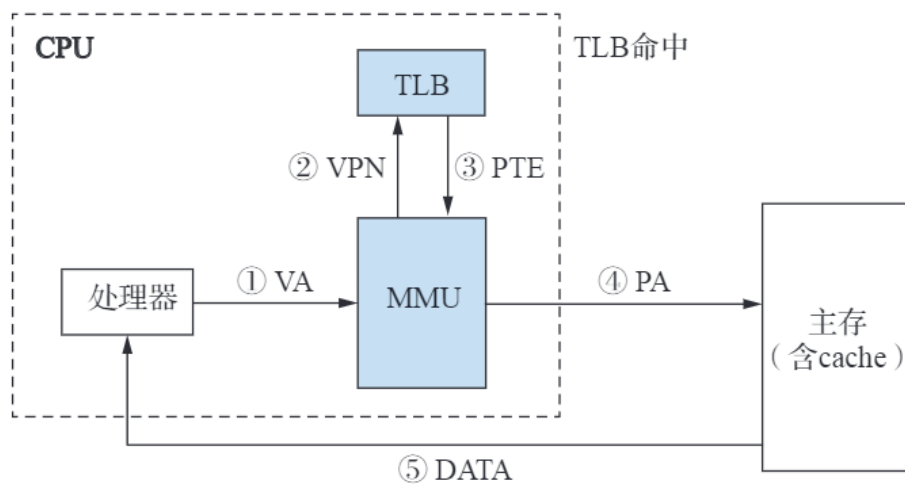


图 4.53 TLB 命中访问流程

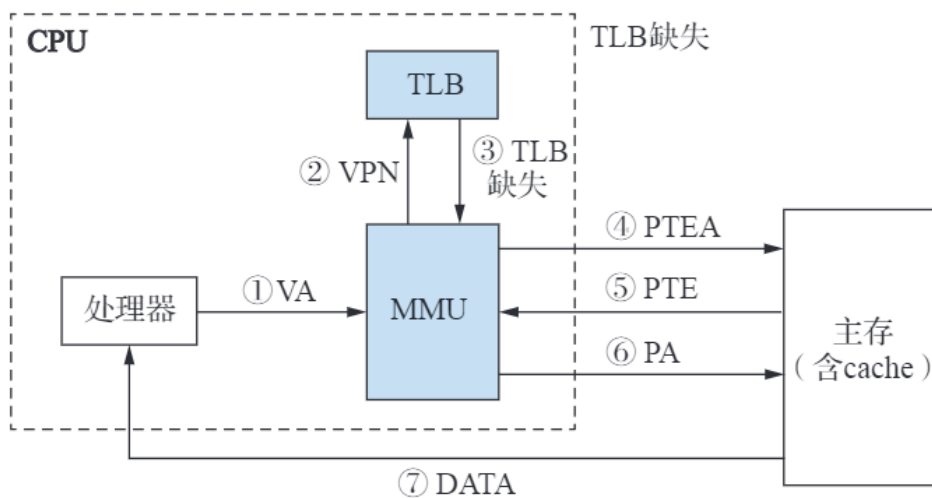


图 4.54 TLB 缺失访问流程

- *TLB* 本质上是小容量的 *Cache* (快速页表)，可以认为其在CPU内部。*TLB* 常采用组相联的方式划分虚拟地址，快速判断要访问的页面是否在内存中。(参照22年计组真题)

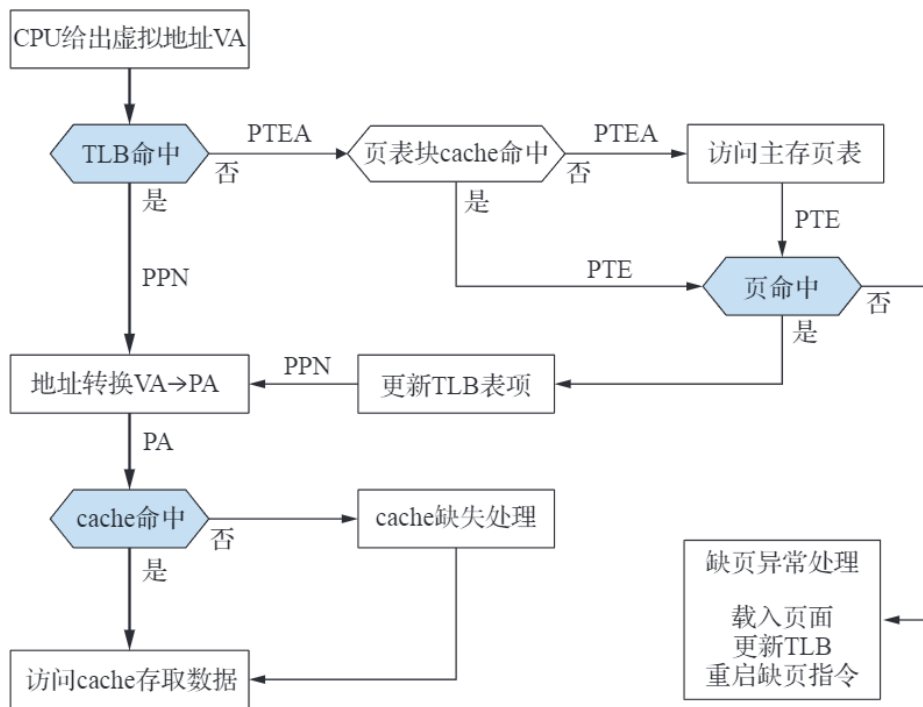


图 4.55 虚存访问操作流程

上图页命中是指 *PTE* 有效位为1。

TLB命中一定页命中

LRU 算法：将近期内最久未被访问的行淘汰