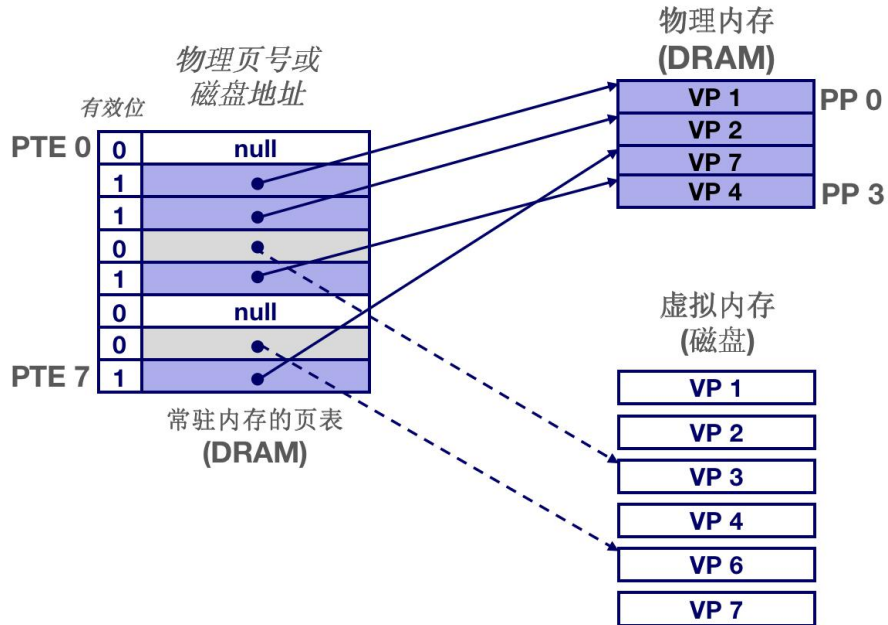


## 第九章知识点

### 1. 虚拟页表和实际物理页的分配关系

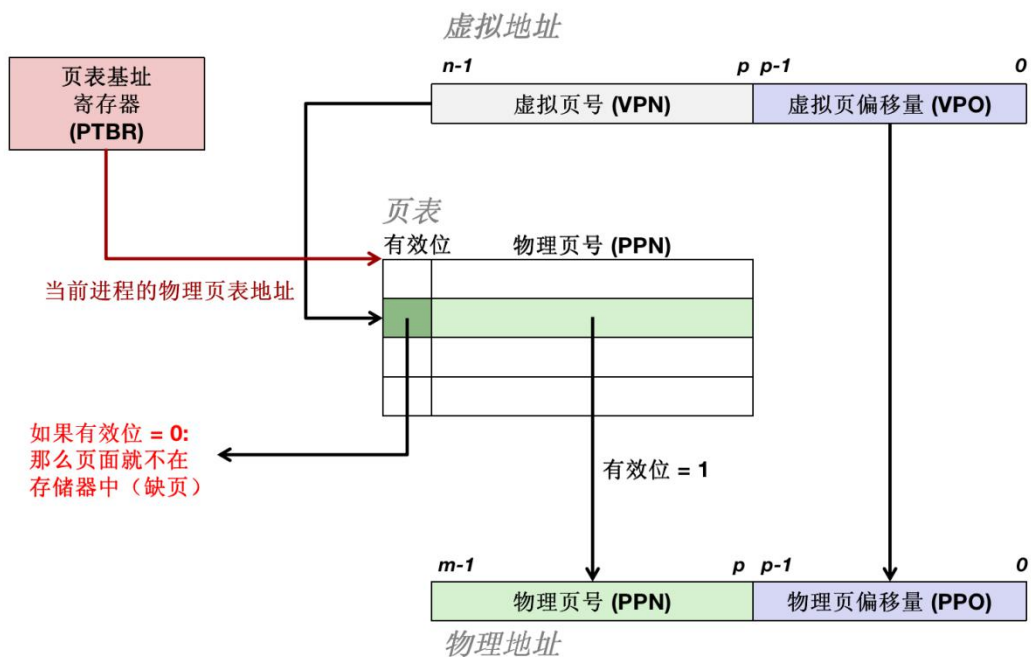


2. 缺页之后的处理方法：选择一个牺牲页，然后进行替换，重新换入相应的页面。然后重新运行触发缺页的指令，此时则可以找到相应的页面。

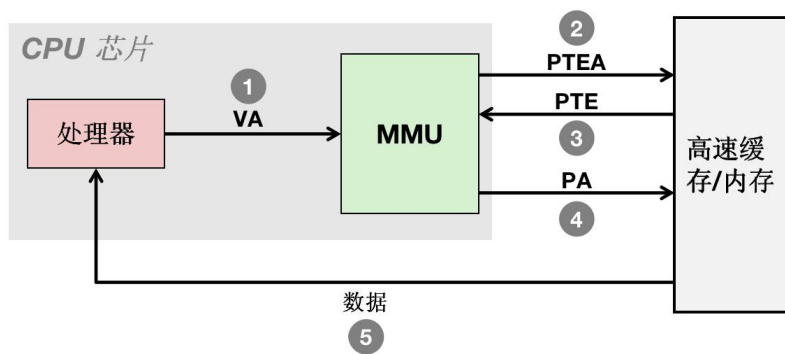
3. 工作集：程序将趋于在一个较小的活动页面集合上工作，这个集合叫做工作集 Working set

如果需要的工作集过大，那么会引起抖动现象。

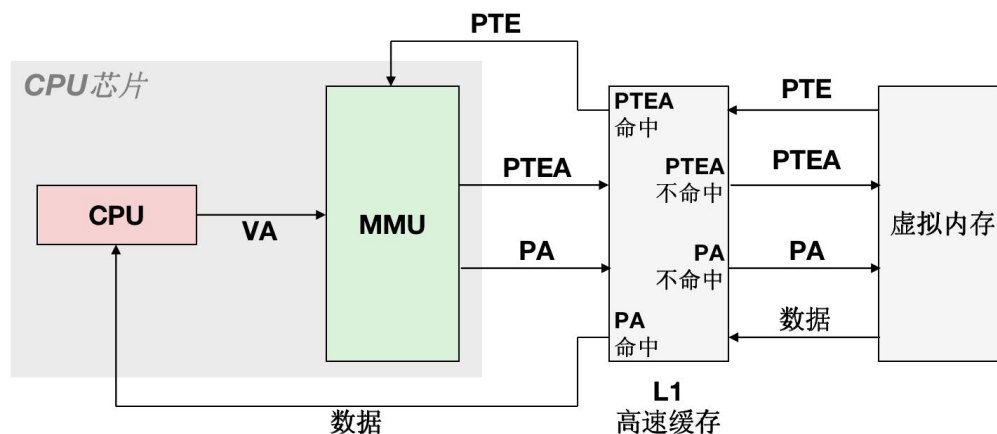
4. 如何进行虚拟地址到实际物理地址的转换。



更 High-Level 层面的转换



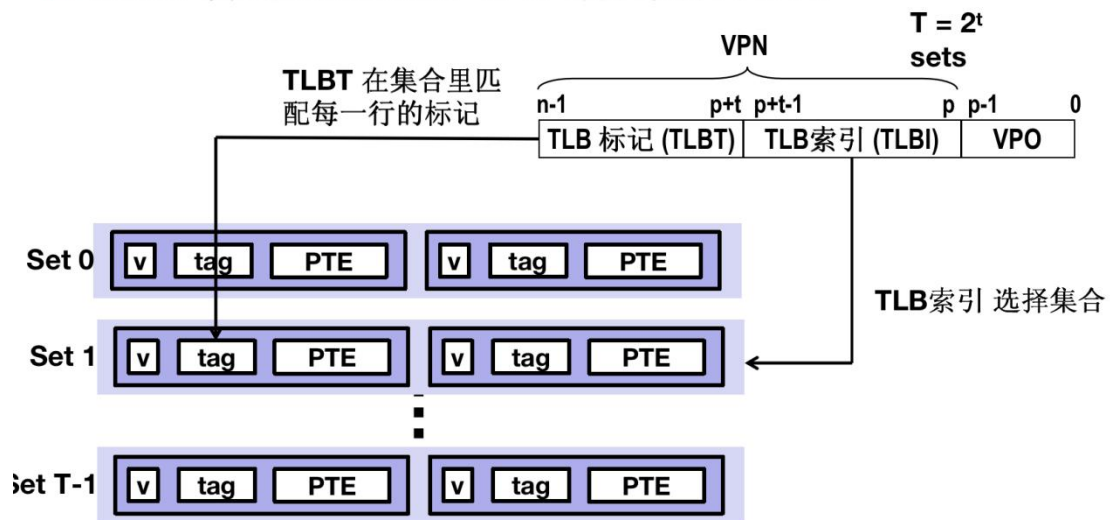
- 1) 处理器生成一个虚拟地址，并将其传送给MMU
- 2-3) MMU 使用内存中的页表生成PTE地址
- 4) MMU 将物理地址传送给高速缓存/主存
- 5) 高速缓存/主存返回所请求的数据字给处理器
5. 虚拟内存和高速缓存结合起来进行数据存取的全过程



**VA: virtual address** 虚拟地址, **PA: physical address** 物理地址,  
**PTE: page table entry** 页表条目, **PTEA = PTE address** 页表条目地址

6. TLB 全称 Translation Lookaside Buffer 快表  
 存储的是多个 PTE 的集合，在 MMU 中。用来加速地址翻译的过程。  
 组织类似于高速缓存的组织，采用多路组相联的形式。  
 TLB 的构成：TLBT + TLBI + VPO

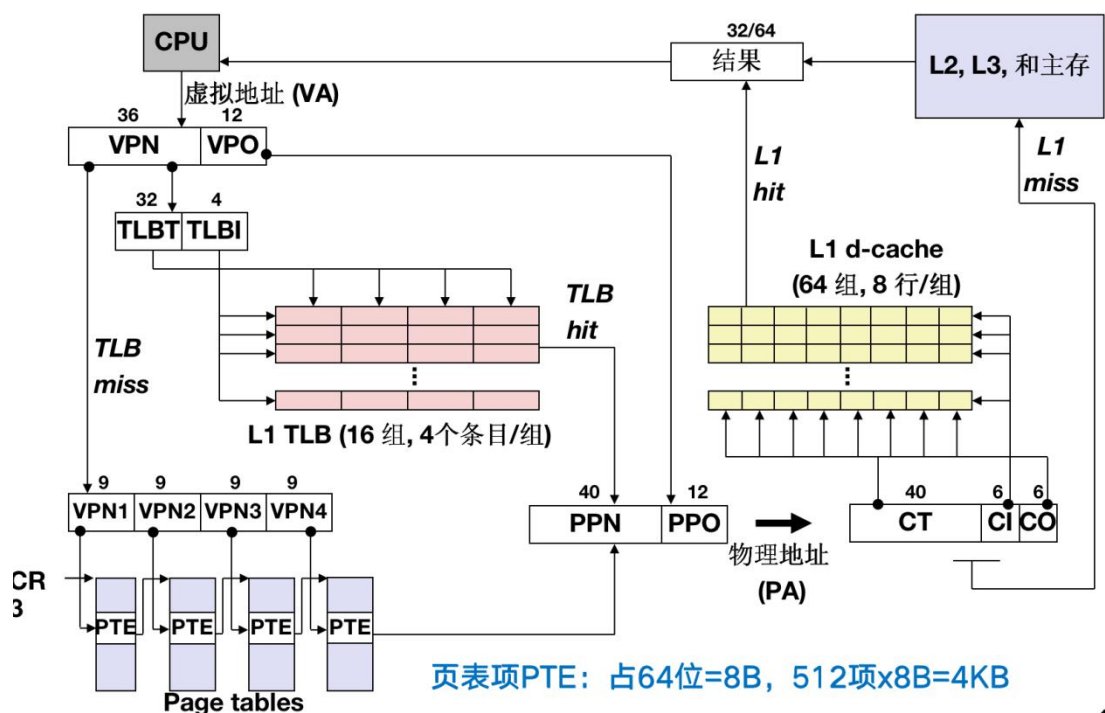
## ■ MMU 使用虚拟地址的 VPN 部分来访问 TLB:



计算 TLB 的平均有效访问时间:  $\text{HitR} * (\text{TLB} + \text{MA}) + (1 - \text{HitR}) * (\text{TLB} + 2 * \text{MA})$

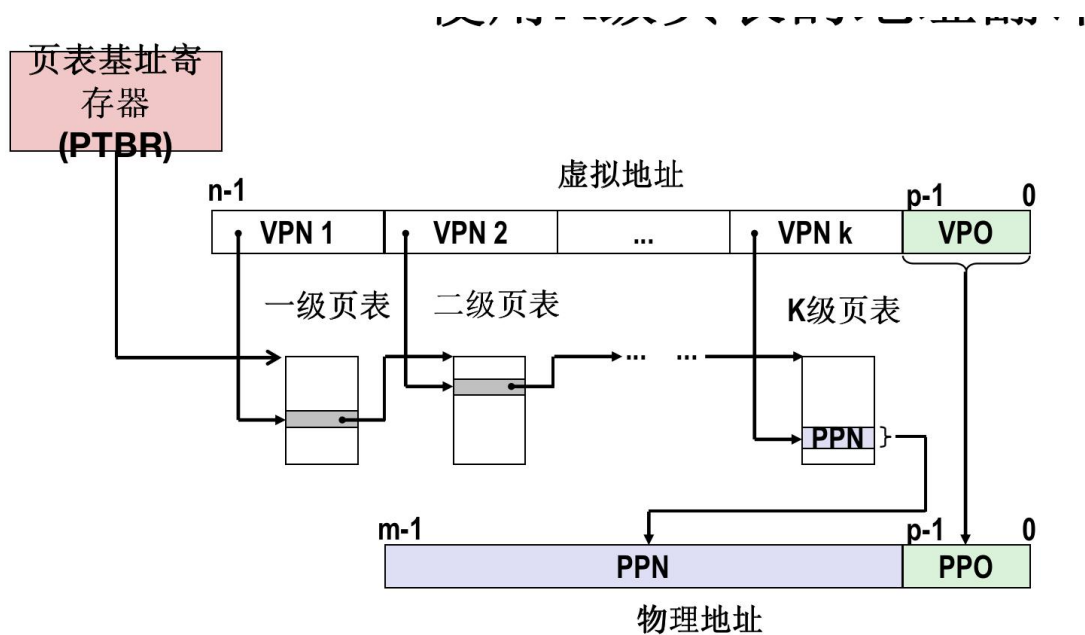
(HitR 代表访问 TLB 的命中率, TLB 代表访问 TLB 的时间, MA 代表访问主存的时间)

引入 TLB 之后, 进行地址转换的整个过程



## 7. 多级页表

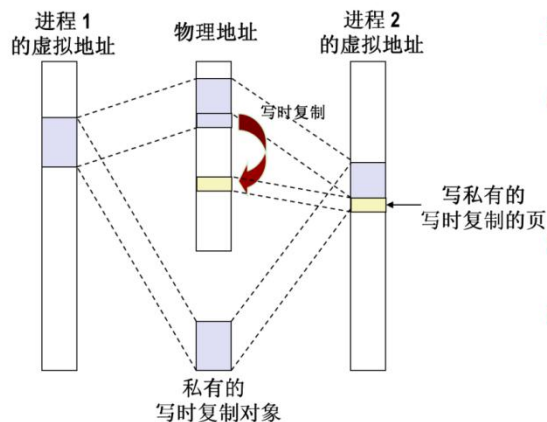
出现的原因: 索引的页表项过多, 导致页表项本身占用的空间过大。



K 级页表的地址转换过程。

## 8. 共享对象带来的 COW (copy-on-write) 机制

### 私有的写时复制 (Copy-on-write) 对象



- 写私有页的指令触发保护故障
- 故障处理程序创建这个页面的一个新副本，更新PTE条目，且可写
- 故障处理程序返回时重新执行写指令
- 尽可能地延迟拷贝（创建副本）充分利用物理内存