



计算机组成原理

第 11 讲

左德承

哈尔滨工业大学计算学部
容错与移动计算研究中心

4.4 虚拟存储器

- 虚拟存储器概述
- 虚拟存储器地址映射与变换
- 举例

虚拟存储系统的基本概念

- 虚拟存储技术的实质

- 程序员在比实际主存空间大得多的逻辑地址空间中编写程序
- 程序执行时，把当前需要的程序段和相应的数据块调入主存，其他暂不用的部分存放在磁盘上
- 指令执行时，通过**硬件**将逻辑地址（也称虚拟地址或虚地址）转化为物理地址（也称主存地址或实地址）
- 在发生程序或数据访问失效(缺页)时，**由操作系统**进行主存和磁盘之间的信息交换
- 虚拟存储器机制由硬件与操作系统共同协作实现，涉及到操作系统中的许多概念，如进程、进程的上下文切换、存储器分配、虚拟地址空间、缺页处理等。

- 页式管理

- 页式虚存把主存与外存均划分成等长的页面。
- 常用页大小为**4KB~64KB**。
- 便于维护，便于生成页表，类似于**cache**的块表
- 不容易产生碎块，存储空间浪费小
- 页不是独立的实体，处理、保护和共享不方便

- 段式管理

- 段页式管理

“Cache—主存”与“主存—辅存”层次的区别 4.4

存储层次 比较项目	“Cache - 主存”层次	“主存 - 辅存”层次
目的	为了弥补主存速度的不足	为了弥补主存容量的不足
存储管理实现	主要由专用硬件实现	主要由软件实现
访问速度的比值 (第一级和第二级)	几比一	几百比一
典型的块(页)大小	几十个字节	几百到几千个字节
CPU对第二级的 访问方式	可直接访问	均通过第一级
失效时CPU是否切换	不切换	切换到其他进程

存储层次中cache和虚存的典型指标

4.4

参数	一级cache	虚拟存储器
块（页）大小	16~128字节	4096~65536字节
命中时间	1~3个时钟周期	50~150个时钟周期
失效开销	8~150个时钟周期	$10^6 \sim 10^7$ 个时钟周期
失效率	0.1%~10%	0.00001%~0.001%
地址映射关系	25~45位物理地址 映射到14~20位 cache地址	32~64位虚地址映射 到25~45位物理地址

有关虚拟存储器的四个问题

◆ 映象规则

全相联。以降低失效率为主要目标。

◆ 查找算法

页表，段表，TLB。

◆ 替换算法

LRU等。

尽可能减少页故障，OS为每个页面设置“使用位”。

◆ 写策略

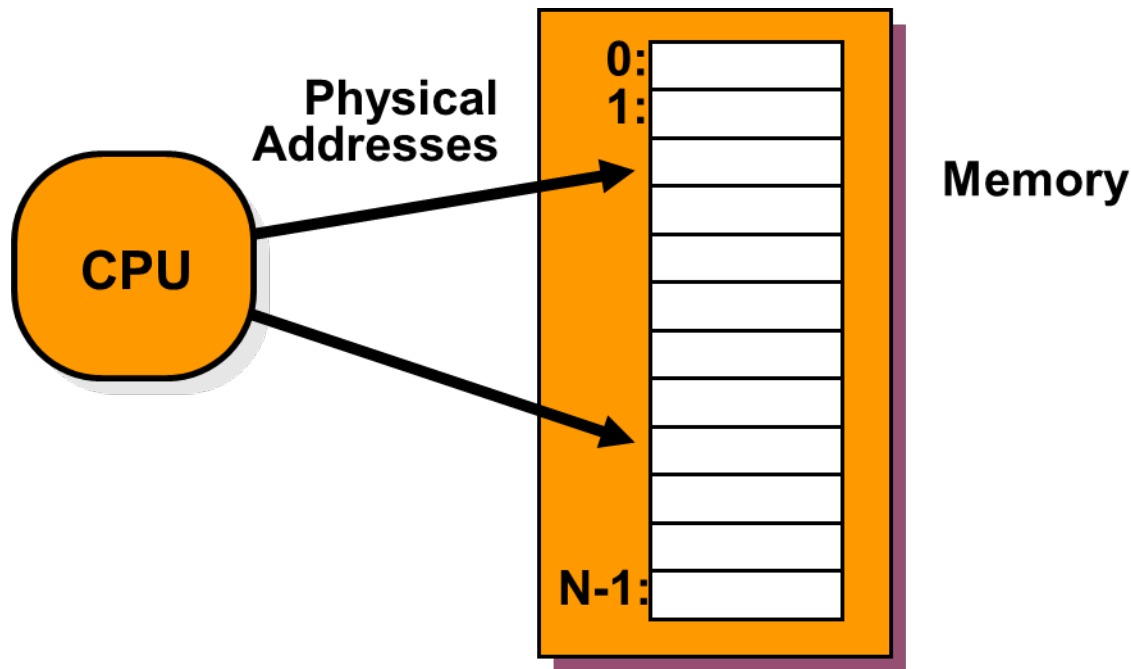
写回法

4.4 虚拟存储器

- 虚拟存储器概述
- 虚拟存储器地址映射与变换
- 举例

实地址计算机系统

- **CPU地址：** 物理内存地址
 - 大多数Cray计算机，早期PC，大多数嵌入式系统

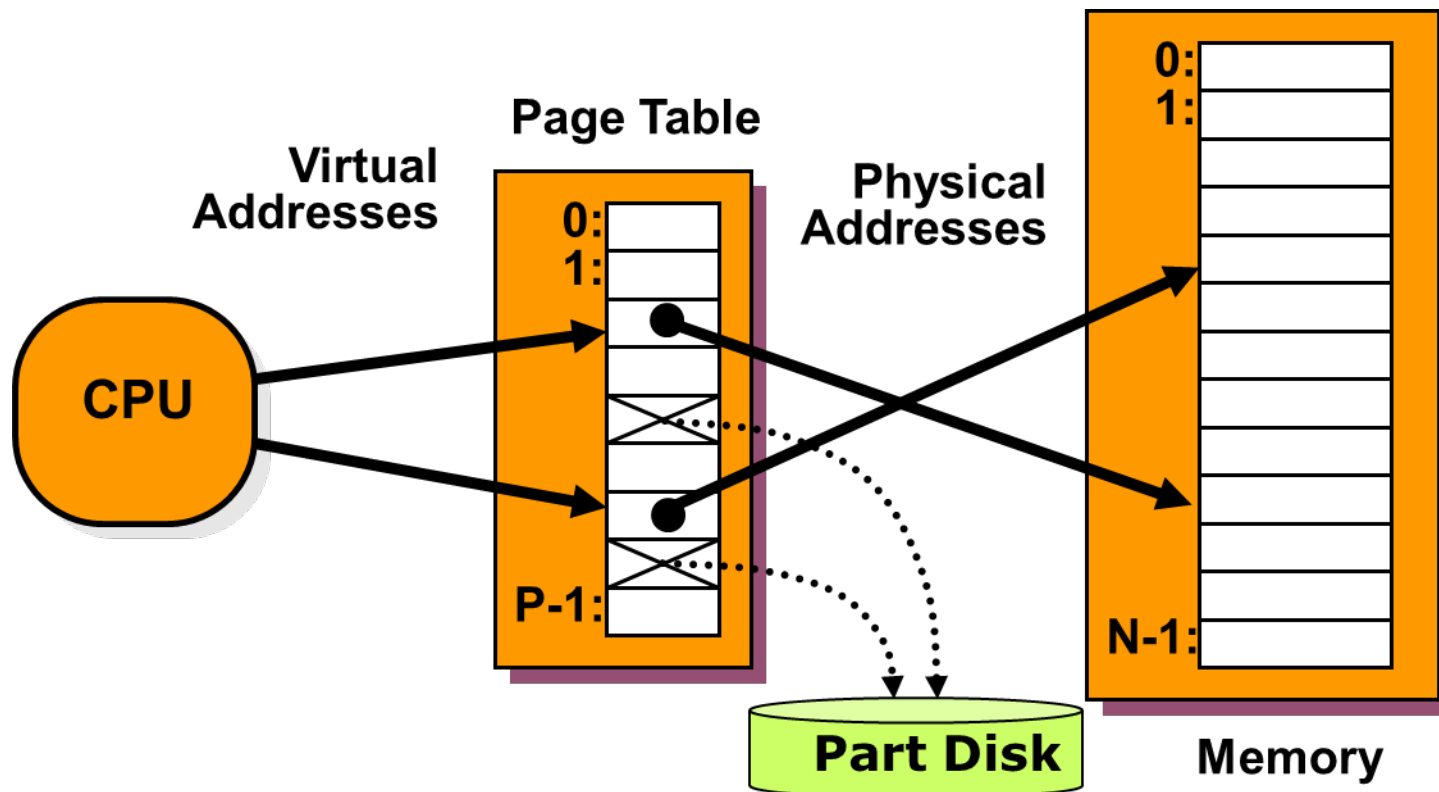


虚地址计算机系统

4.4

- **CPU地址：虚拟地址**

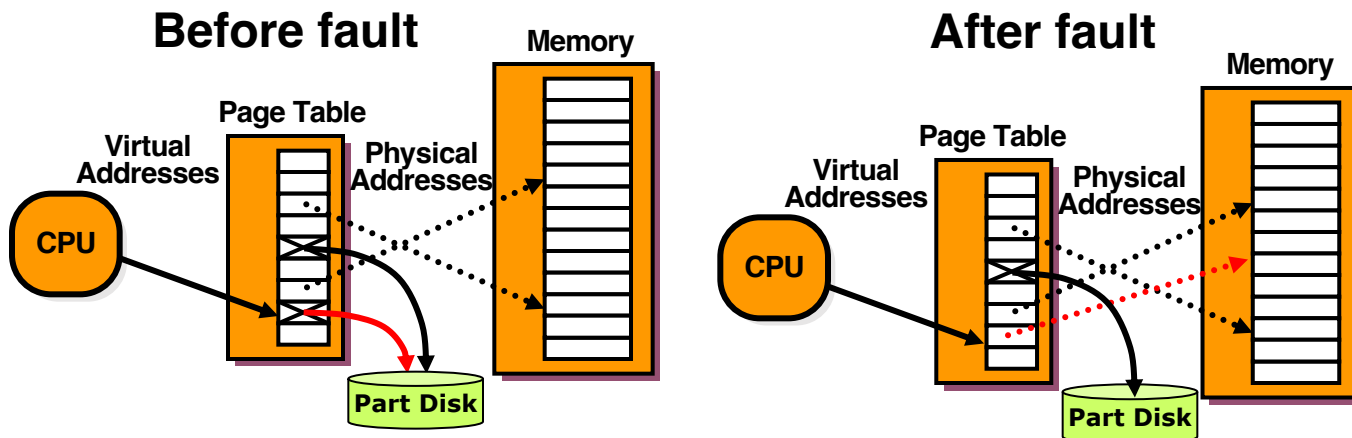
- 需要虚实变换，硬件通过OS维护的页表将虚拟地址转换为物理地址
- 工作站，服务器，现代PC



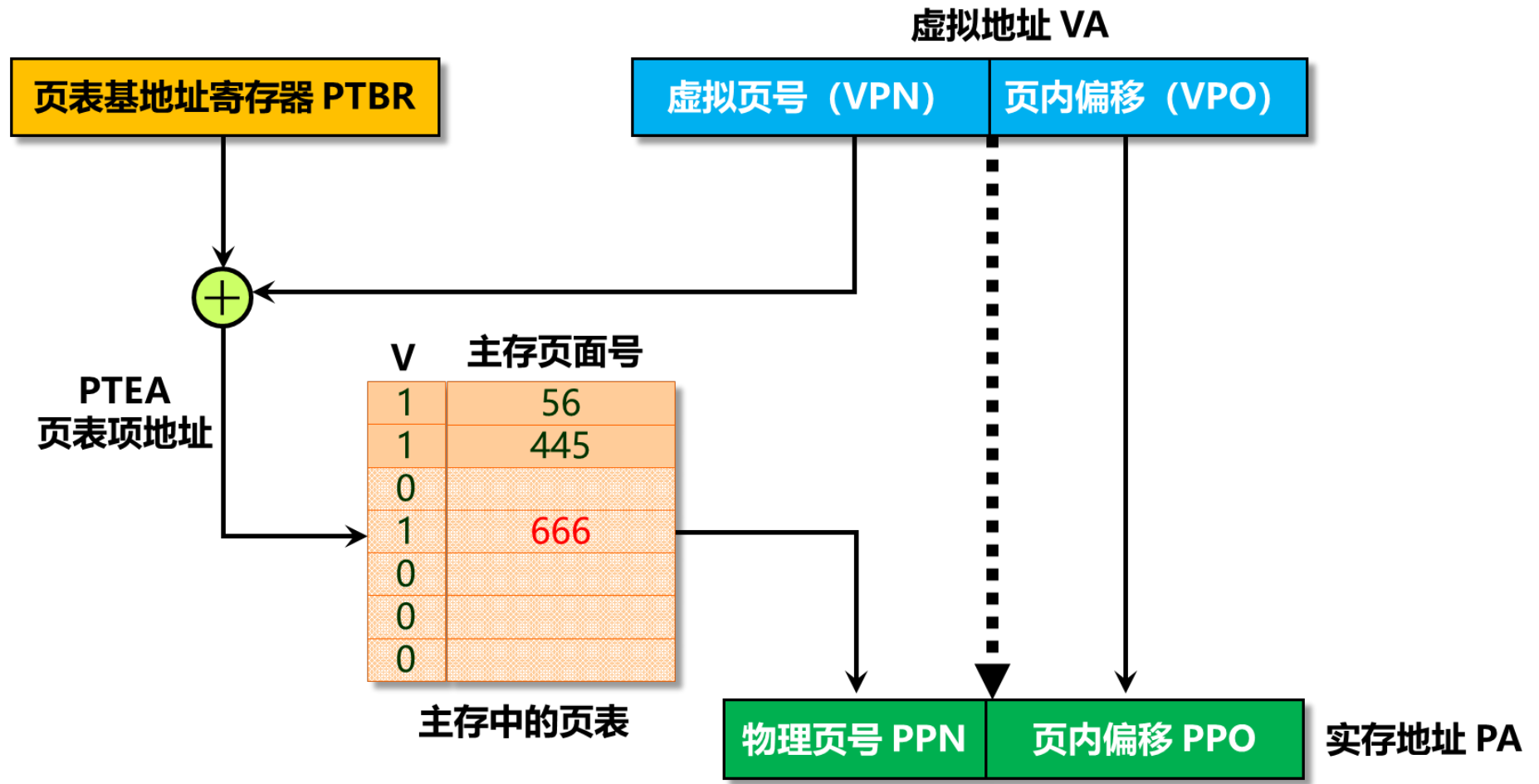
缺页 Page Faults

4.4

- 页表指示虚拟地址不在内存中
 - 操作系统负责将数据从磁盘迁移到内存中
 - 当前进程挂起
 - 操作系统负责所有的替换策略
 - 唤醒挂起进程



页式虚拟存储器结构



◆ 页表首址记录在页表基址寄存器中

页表首地址



	装入位	修改位	替换控制位	其他	实页号 (8 进制)
0 虚页	1				11
1 虚页	1				13
2 虚页	1				16
3 虚页	1				10
4 虚页	1				14

- 每个进程有一个页表，其中有装入位、修改（Dirt）位、替换控制位、访问权限位、禁止缓存位、实页号。

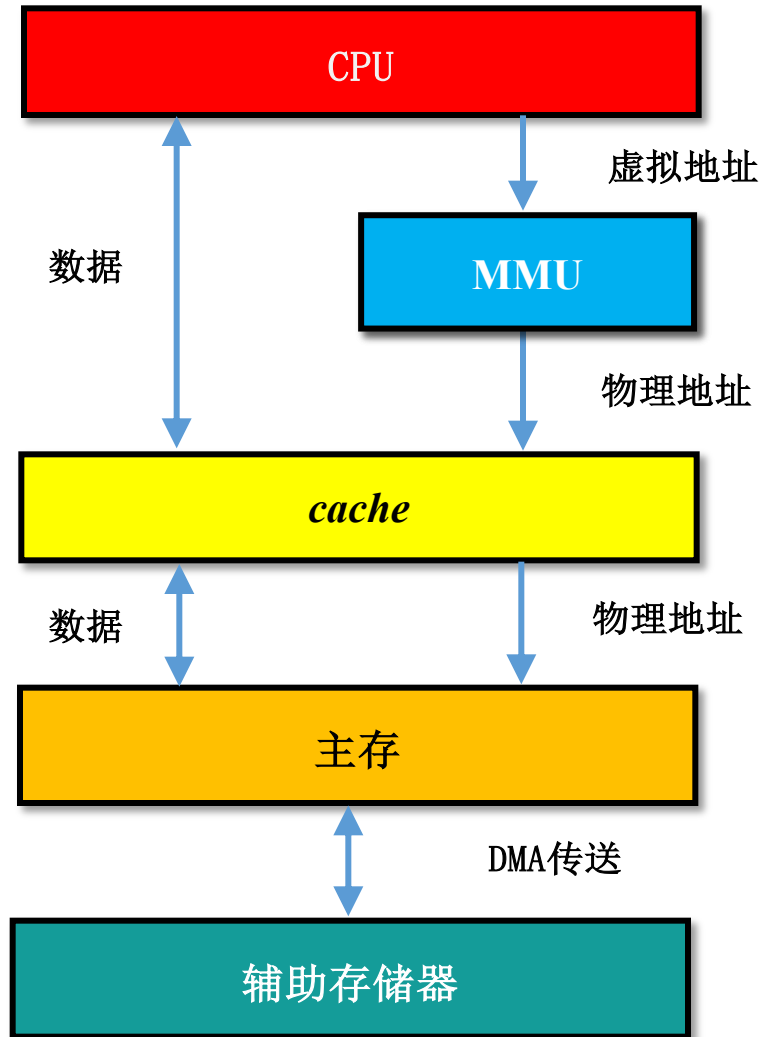
- 一个页表的项数由什么决定？

理论上由虚拟地址空间大小决定。

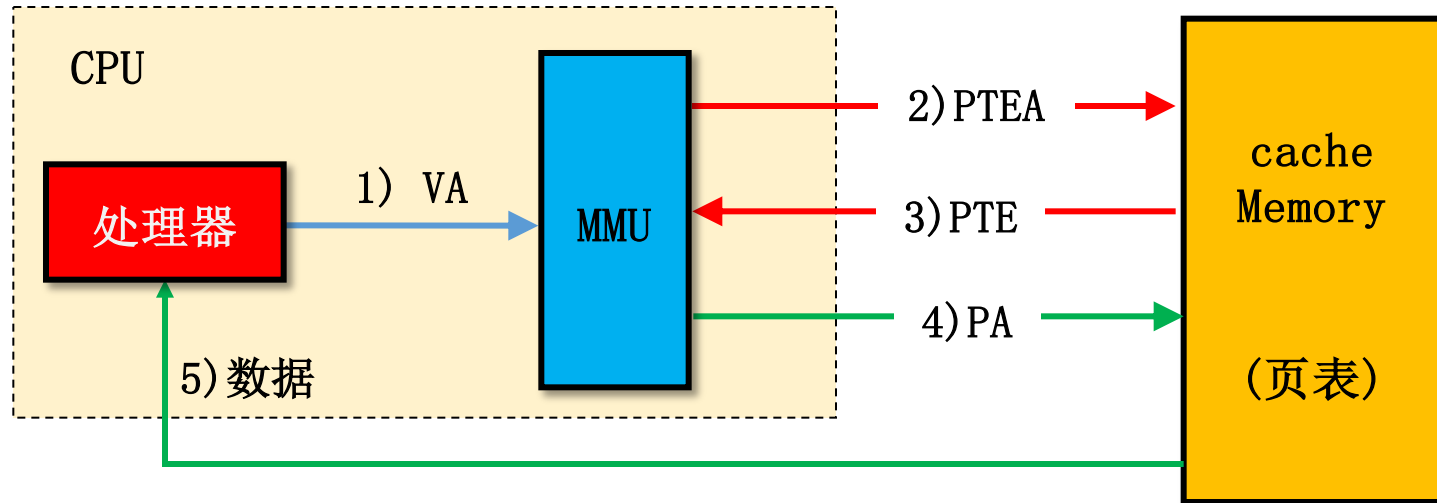
- 每个进程的页表大小一样吗？

各进程有相同虚拟空间，故理论上一样。实际大小看具体实现方式，如“空洞”页面如何处理等

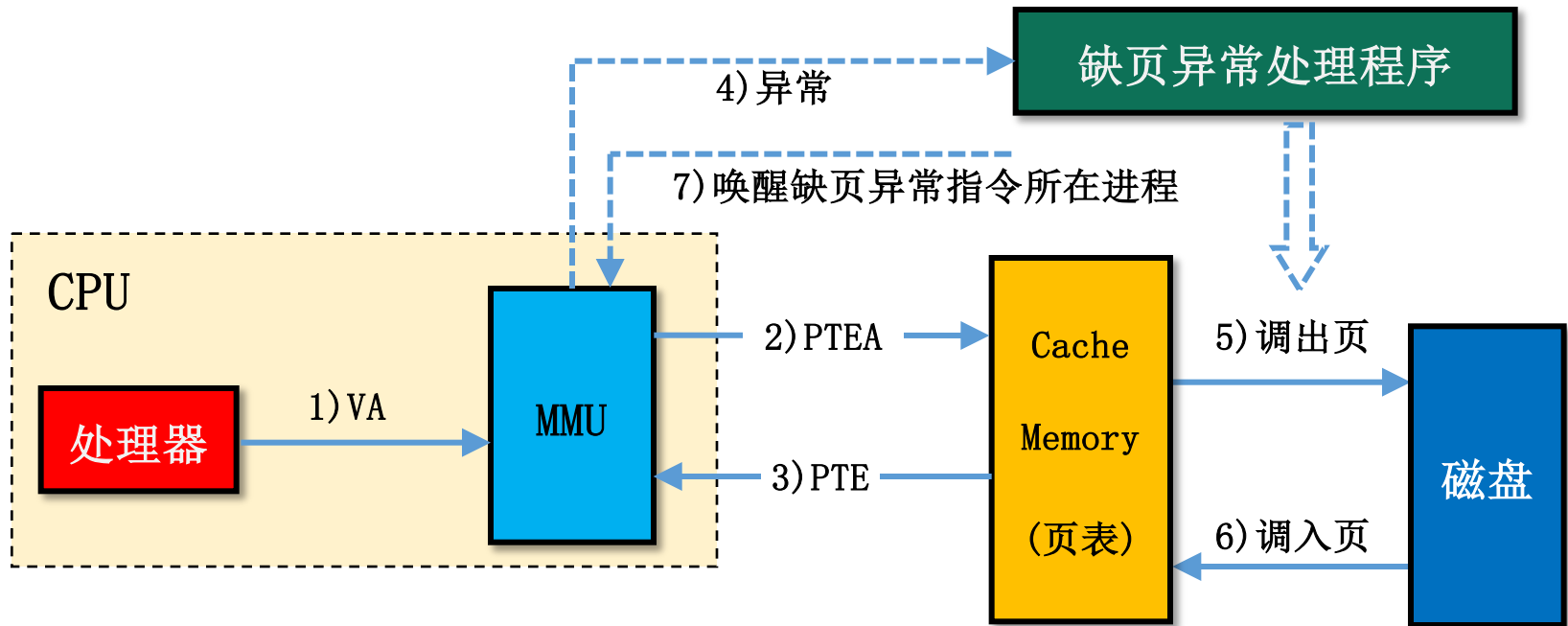
层次性结构



虚拟地址→物理地址（页命中）



虚拟地址→物理地址（缺页）



快表提高地址转换速度

- 地址转换速度慢

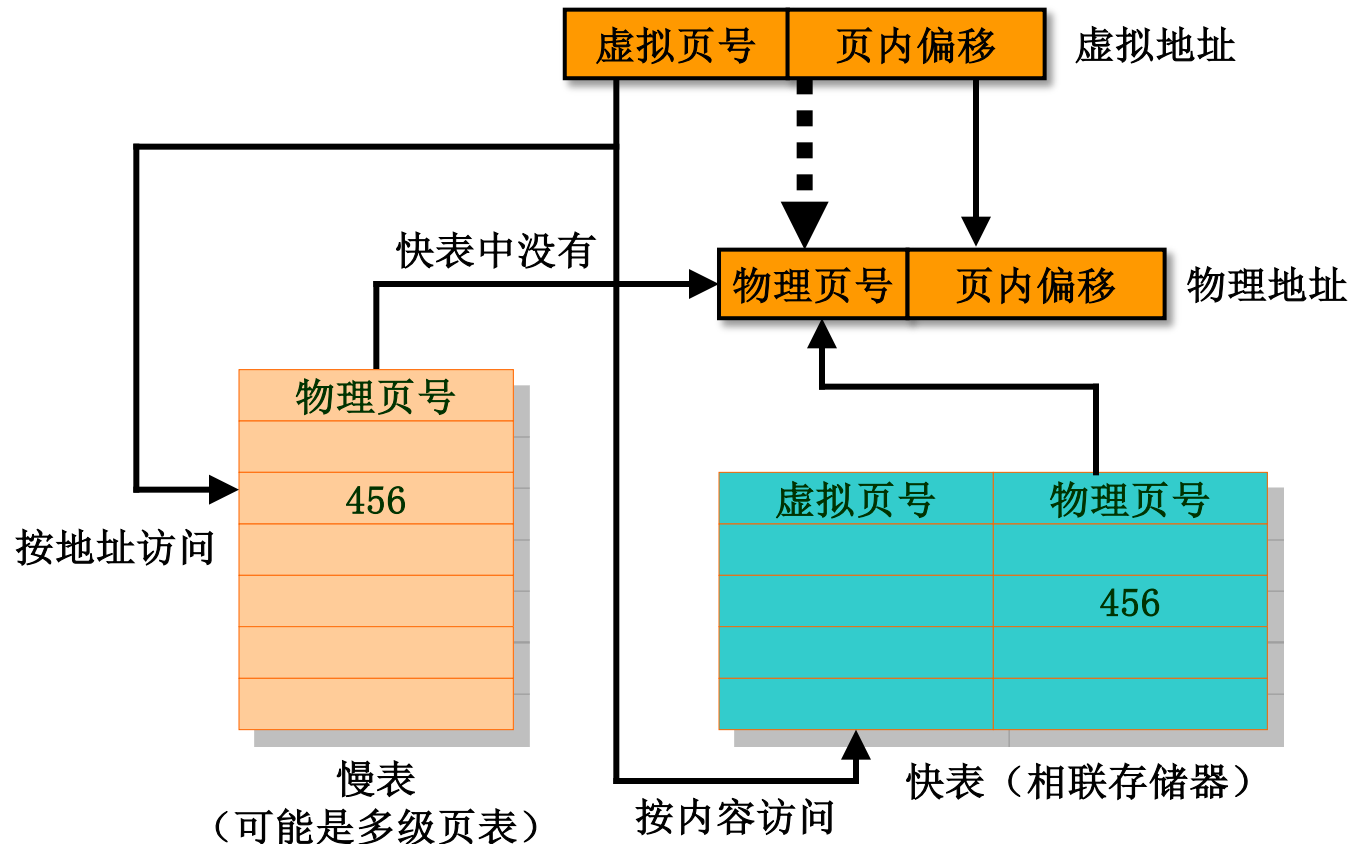
- 访问页表，访问数据，需2次访存，速度慢
- 为缩小页表大小，OS普遍采用多级页表结构，速度更慢

- 加速方法：

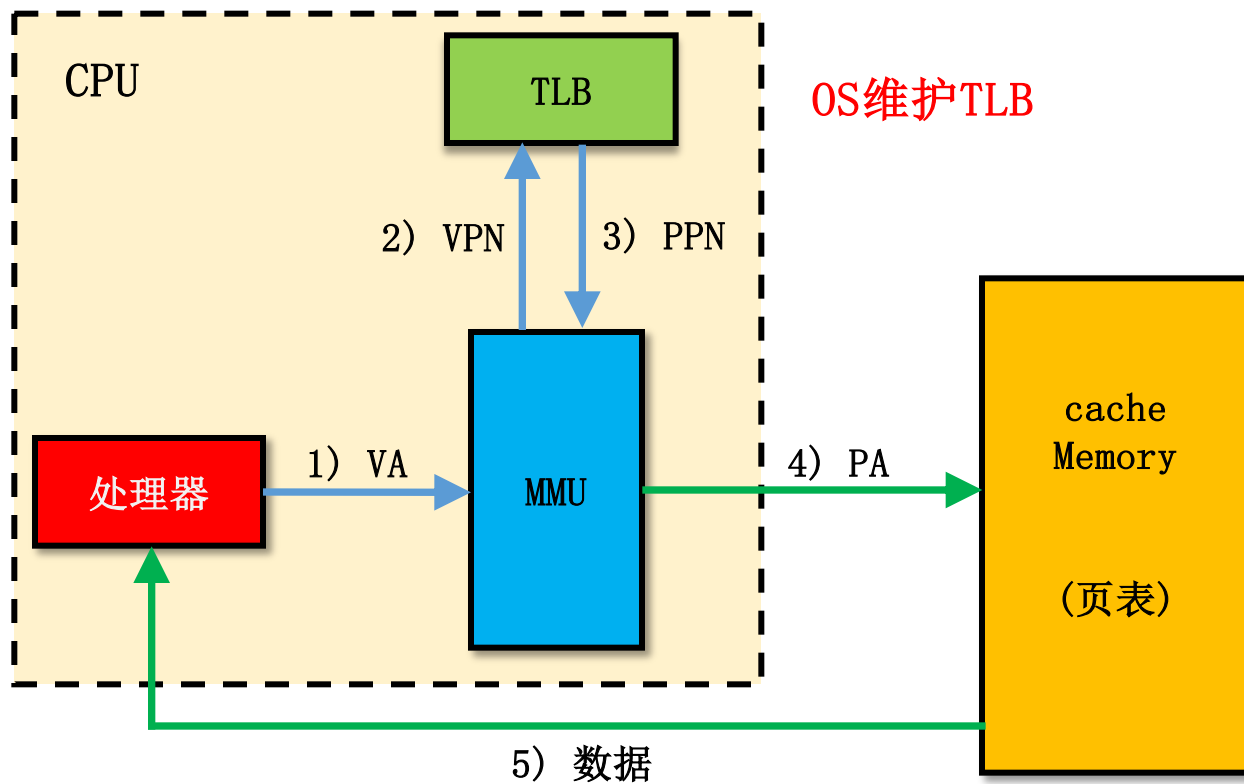
引入一个体积小的快表TLB（ Translation Look-Aside Buffers ）

- 缓存页表中经常被访问的表项
- (Valid, VPN, PPN)
- 快表引入相联存储器机制，提高查找速度
 - 采用随机替换算法

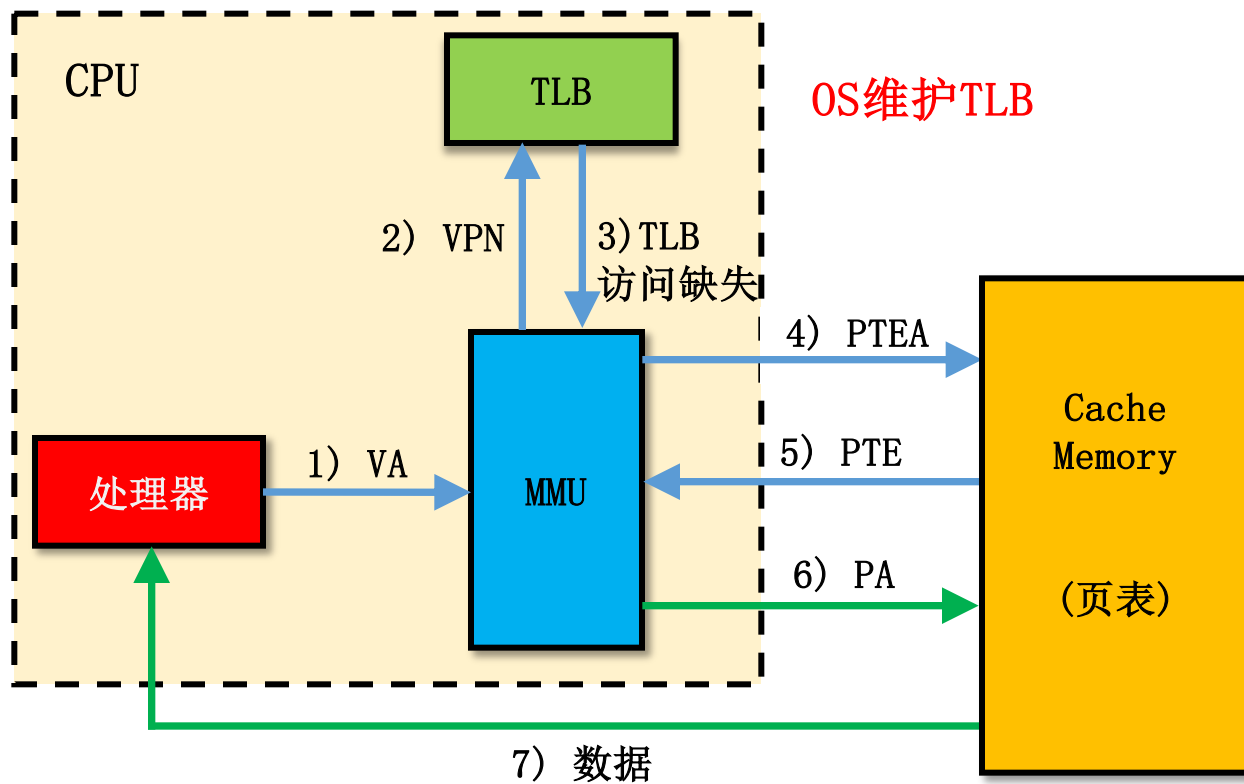
经快慢表实现内部地址转换



TLB 命中

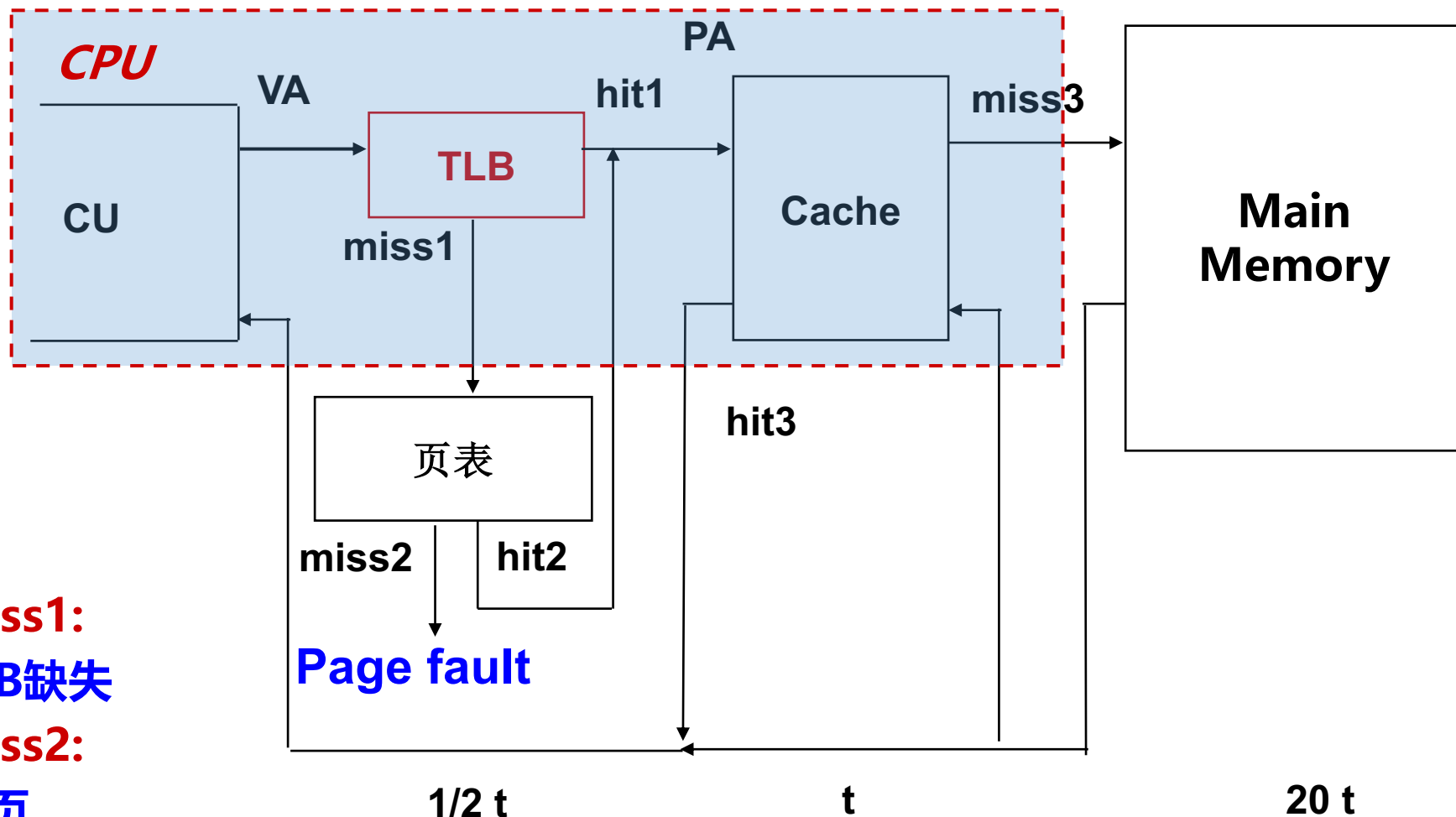


TLB 缺失



Translation Look-Aside Buffers

4.4



Miss1:
TLB缺失

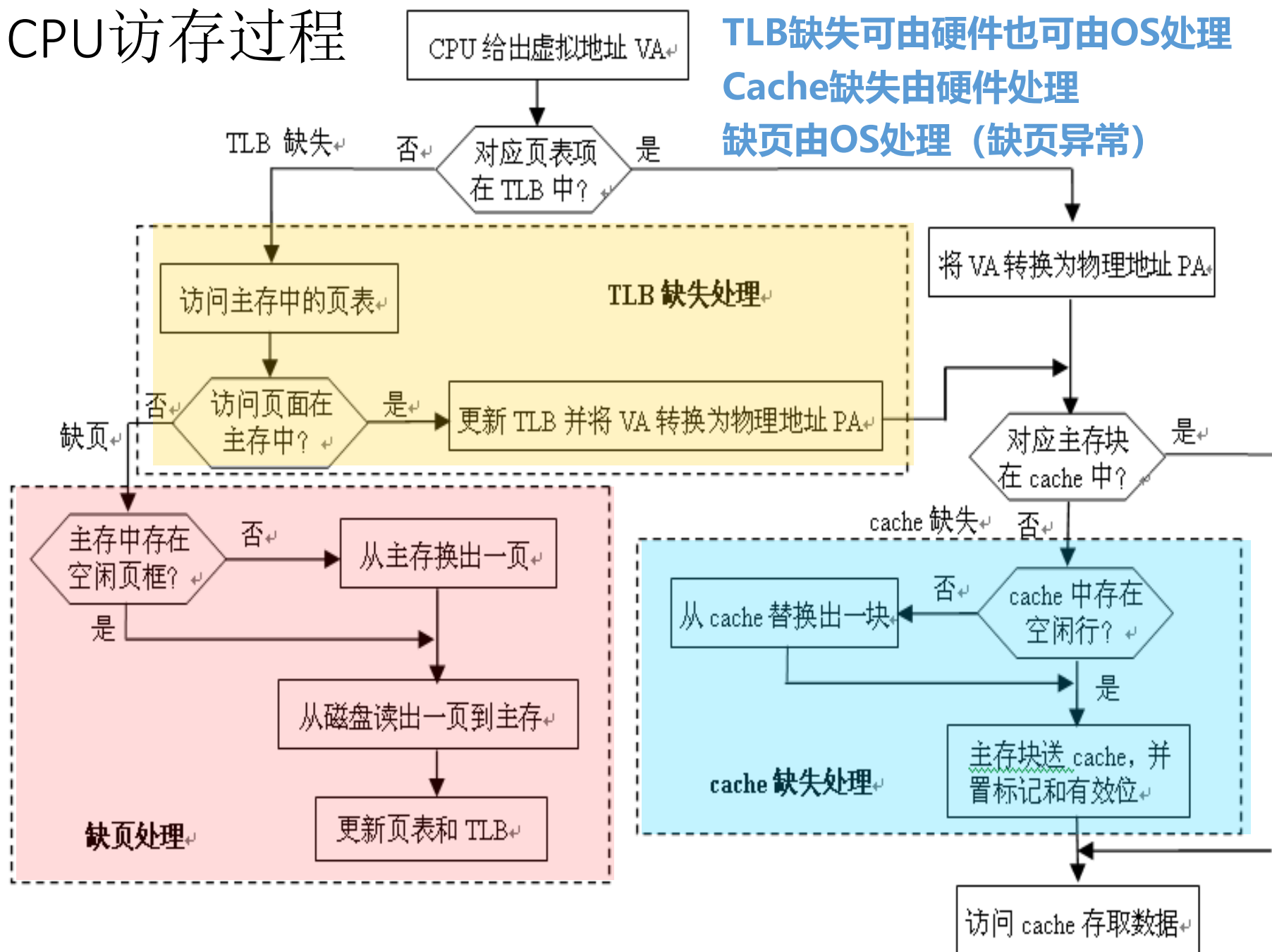
Miss2:
缺页

Miss3:
PA 在主存中，但不在Cache中

TLB冲刷指令和Cache冲刷指令
都是操作系统使用的特权指令

CPU访存过程

TLB缺失可由硬件也可由OS处理
Cache缺失由硬件处理
缺页由OS处理（缺页异常）



举例：三种不同缺失的组合

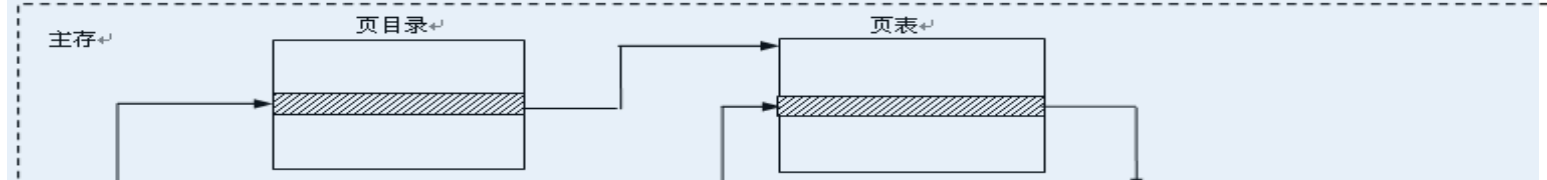
TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	可能，TLB命中则页表一定命中，但实际上不会查页表
miss	hit	hit	可能，TLB缺失但页表命中，信息在主存，就可能在Cache
miss	hit	miss	可能，TLB缺失但页表命中，信息在主存，但可能不在Cache
miss	miss	miss	可能，TLB缺失页表缺失，信息不在主存，一定也不在Cache
hit	miss	miss	不可能，页表缺失，信息不在主存，TLB中一定没有该页表项
hit	miss	hit	同上
miss	miss	hit	不可能，页表缺失，信息不在主存，Cache中一定也无该信息

最好的情况是hit、hit、hit，此时，访问主存几次？ 不需要访问主存！

以上组合中，最好的情况是？ hit、hit、miss和miss、hit、hit 访存1次

以上组合中，最坏的情况是？ miss、miss、miss 需访问磁盘、并访存至少2次

介于最坏和最好之间的是？ miss、hit、miss 不需访问磁盘、但访存至少2次



虚拟地址

TLB

页表

物理地址

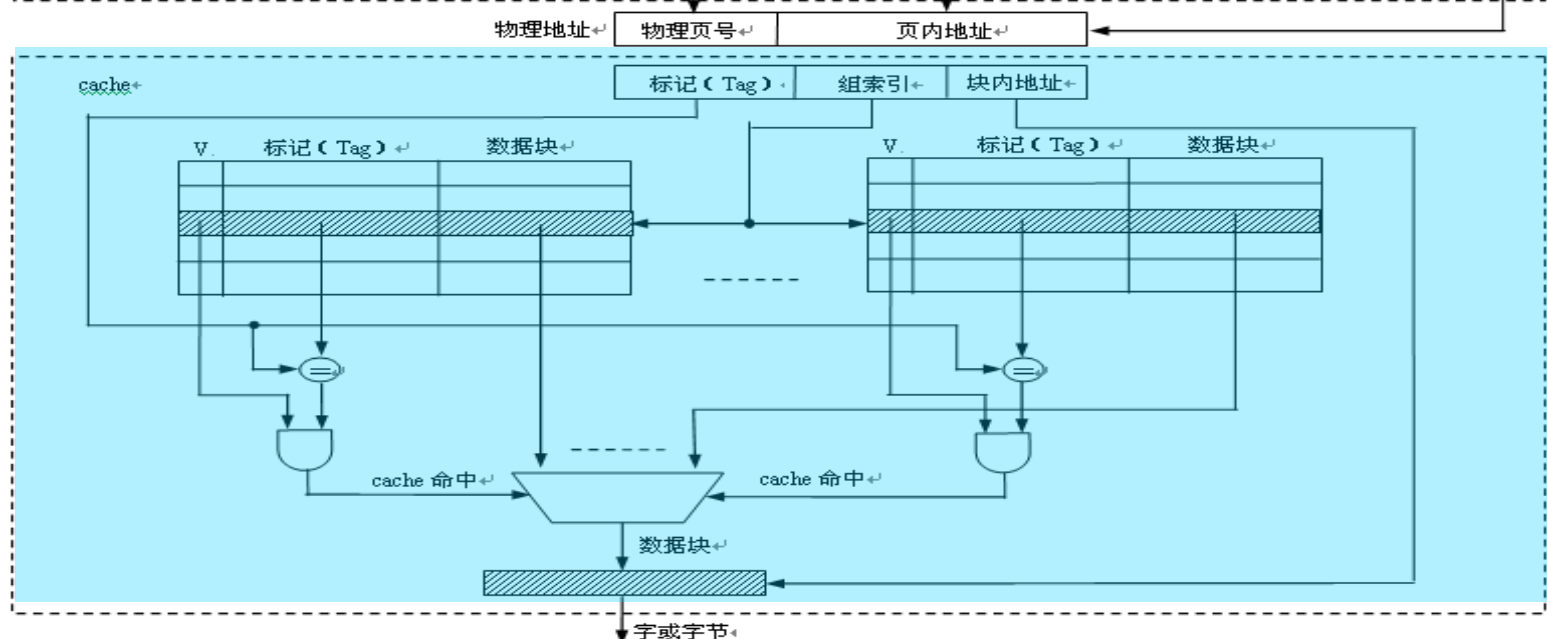
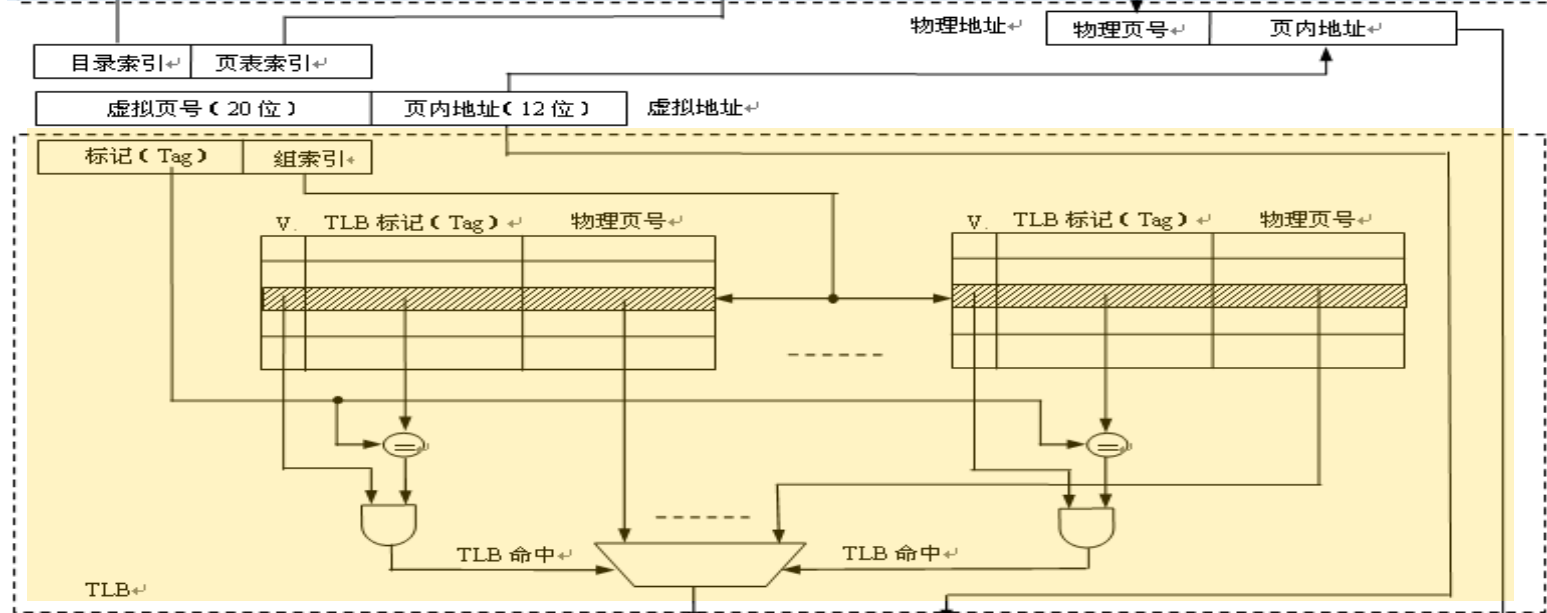
缺页处理

命中

cache

缺失

主存



4.4 虚拟存储器

- 虚拟存储器概述
- 虚拟存储器地址映射与变换
- 举例

缩写的含义

- 基本参数 (按字节编址)
 - $N = 2^n$: 虚拟地址空间大小
 - $M = 2^m$: 物理地址空间大小
 - $P = 2^p$: 页大小
- 虚拟地址 (VA) 中的各字段
 - TLBI: TLB index (TLB索引)
 - TLBT: TLB tag (TLB标记)
 - VPO: Virtual page offset (页内偏移地址)
 - VPN: Virtual page number (虚拟页号)
- 物理地址(PA)中的各字段
 - PPO: Physical page offset (页内偏移地址)
 - PPN: Physical page number (物理页号)
 - CO: Byte offset within cache line (块内偏移地址)
 - CI: Cache index (cache索引)
 - CT: Cache tag (cache标记)

一个简化的存储系统举例

• 假定以下参数，则虚拟地址和物理地址如何划分？共多少页表项？

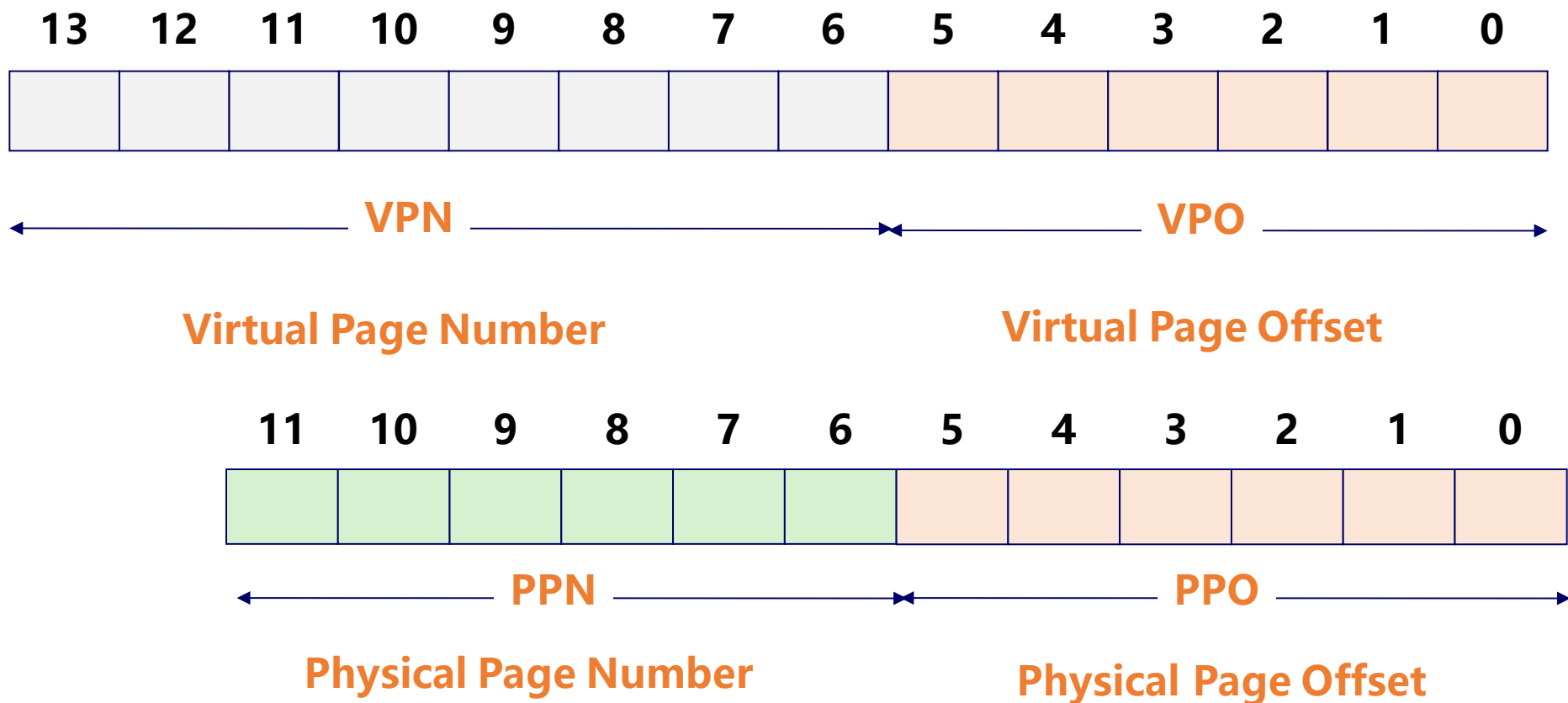
• 14-bit virtual addresses (虚拟地址14位)

• 12-bit physical address (物理地址12位)

• Page size = 64 bytes (页大小64B)

页表项数应为：

$$2^{14-6} = 256$$



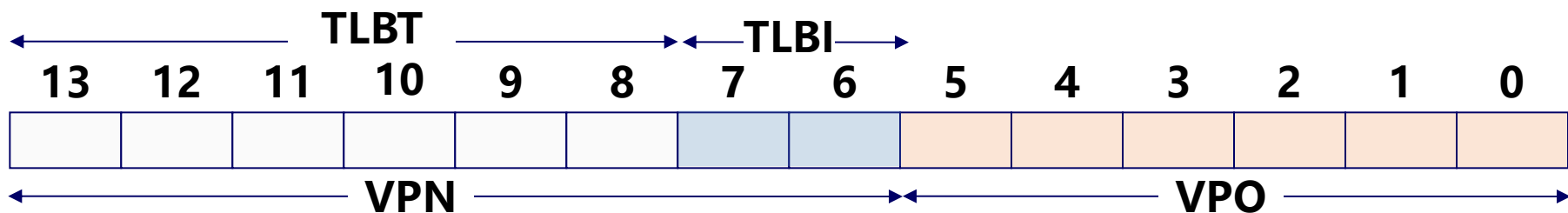
一个简化的存储系统举例（续）

假定部分页表项内容（十六进制表示）如右：

<i>VPN</i>	<i>PPN</i>	<i>Valid</i>
000	28	1
001	—	0
002	33	1
003	02	1
004	—	0
005	16	1
006	—	0
007	—	0

<i>VPN</i>	<i>PPN</i>	<i>Valid</i>
028	13	1
029	17	1
02A	09	1
02B	—	0
02C	—	0
02D	2D	1
02E	11	1
02F	0D	1

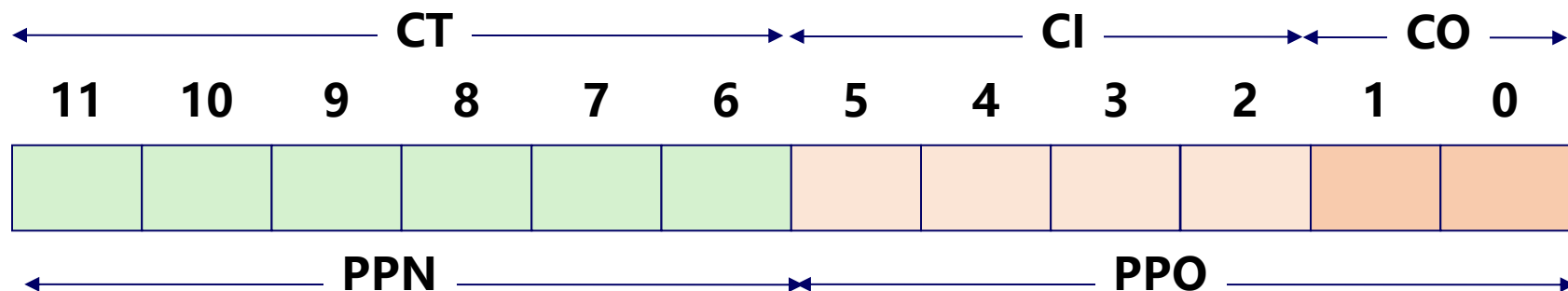
假定TLB如下：16个TLB项，4路组相联，则TLBT和TLBI各占几位？



<i>Set</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>
0	03	—	0	09	0D	1	00	—	0	07	02	1
1	03	2D	1	02	—	0	04	—	0	0A	—	0
2	02	—	0	08	—	0	06	—	0	03	—	0
3	07	—	0	03	0D	1	0A	34	1	02	—	0

一个简化的存储系统举例（续）

假定Cache的参数和内容（十六进制）如下：**16行**，**主存块大小为4B**，**直接映射**，则**主存地址如何划分**？

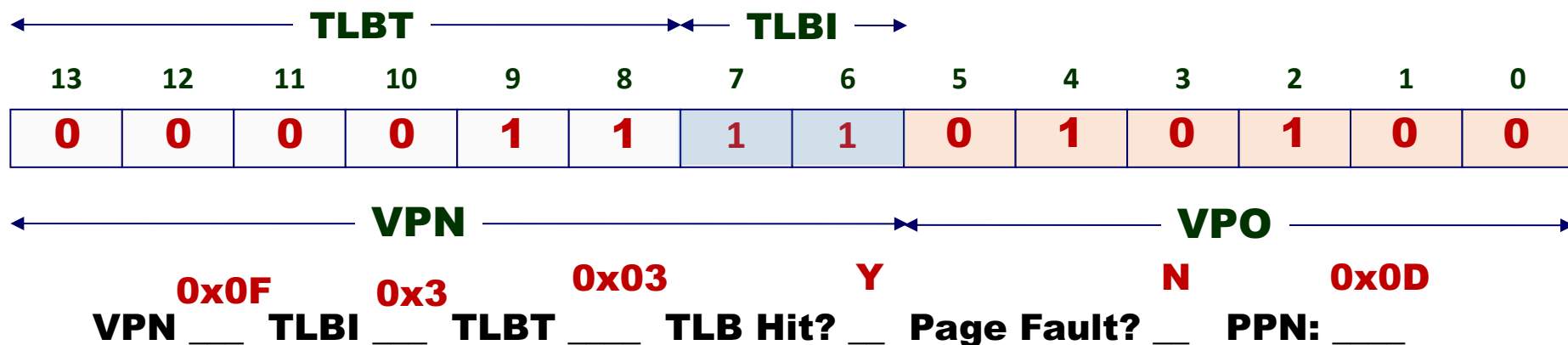


<i>Idx</i>	<i>Tag</i>	<i>V</i>	<i>B0</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>
0	19	1	99	11	23	11
1	15	0	—	—	—	—
2	1B	1	00	02	04	08
3	36	0	—	—	—	—
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	—	—	—	—
7	16	1	11	C2	DF	03

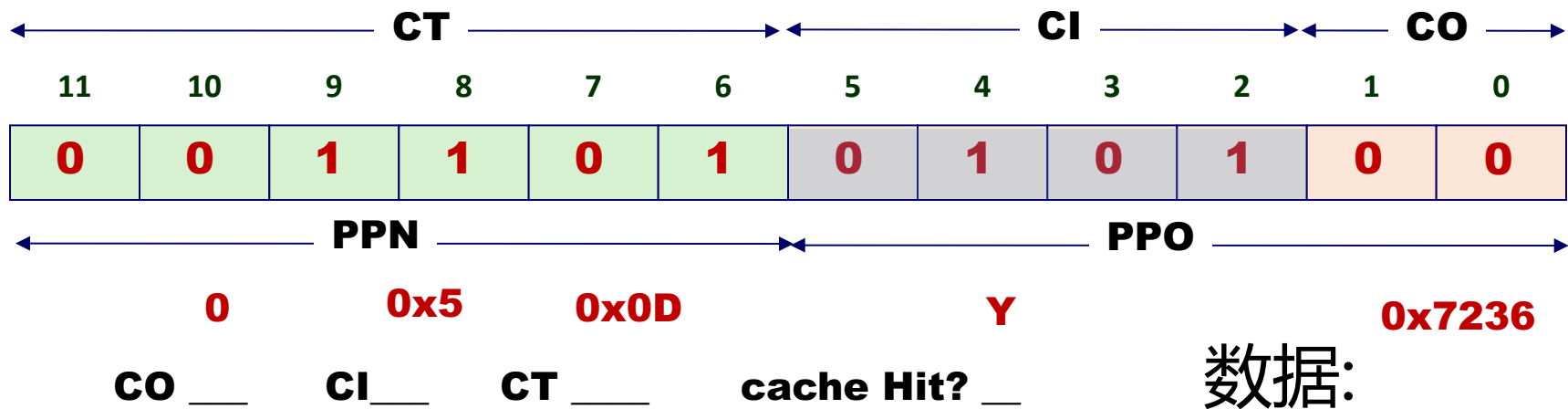
<i>Idx</i>	<i>Tag</i>	<i>V</i>	<i>B0</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>
8	24	1	3A	00	51	89
9	2D	0	—	—	—	—
A	2D	1	93	15	DA	3B
B	0B	0	—	—	—	—
C	12	0	—	—	—	—
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	—	—	—	—

一个简化的存储系统举例（续）

假设该存储系统所在计算机采用小端方式，CPU执行某指令过程中要求访问一个16位数据，给出的逻辑地址为0x03D4，说明访存过程。



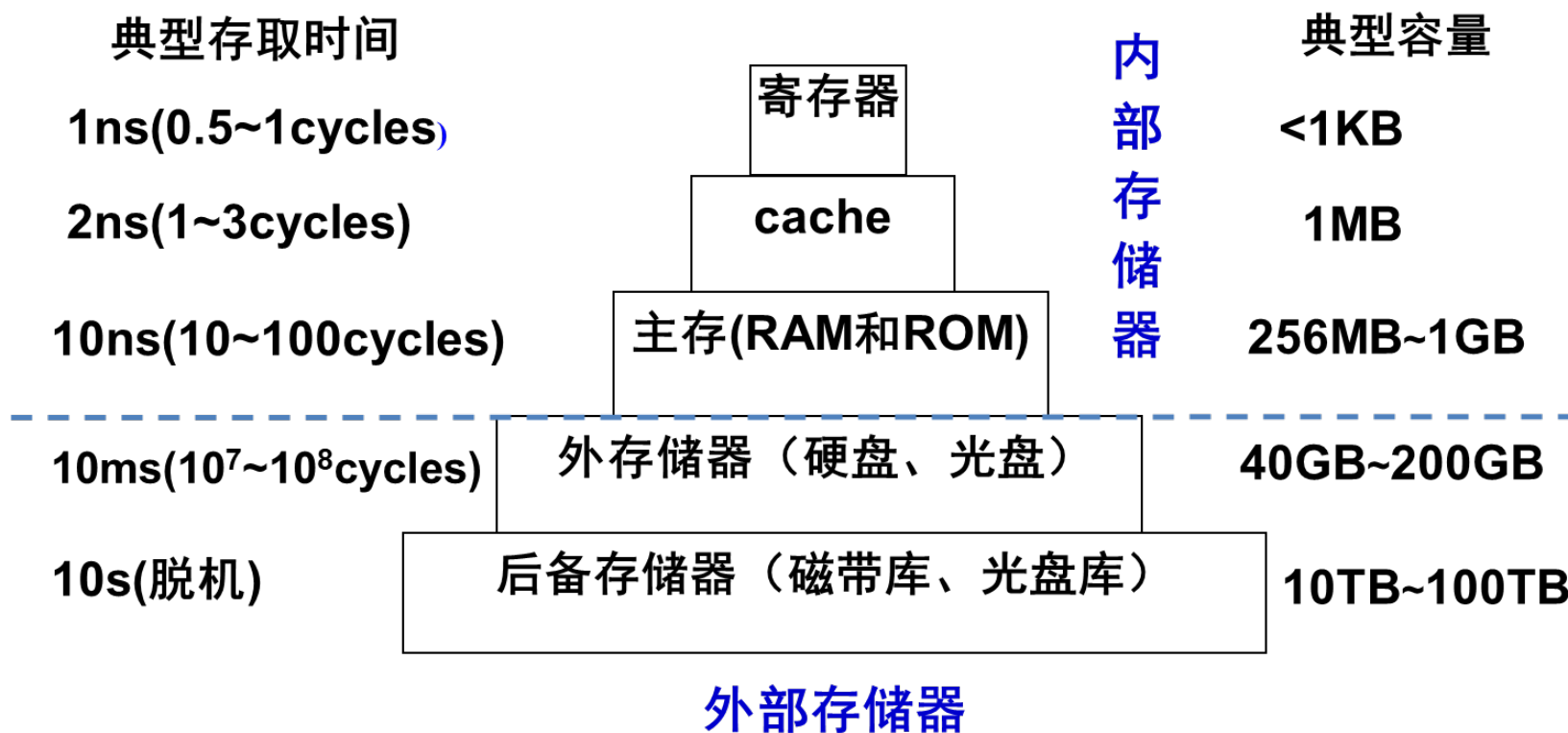
物理地址为 **问题：逻辑地址为0x0A7A、0x0507时的访存过程如何？**
TLB缺失/cache缺失、TLB缺失/缺页



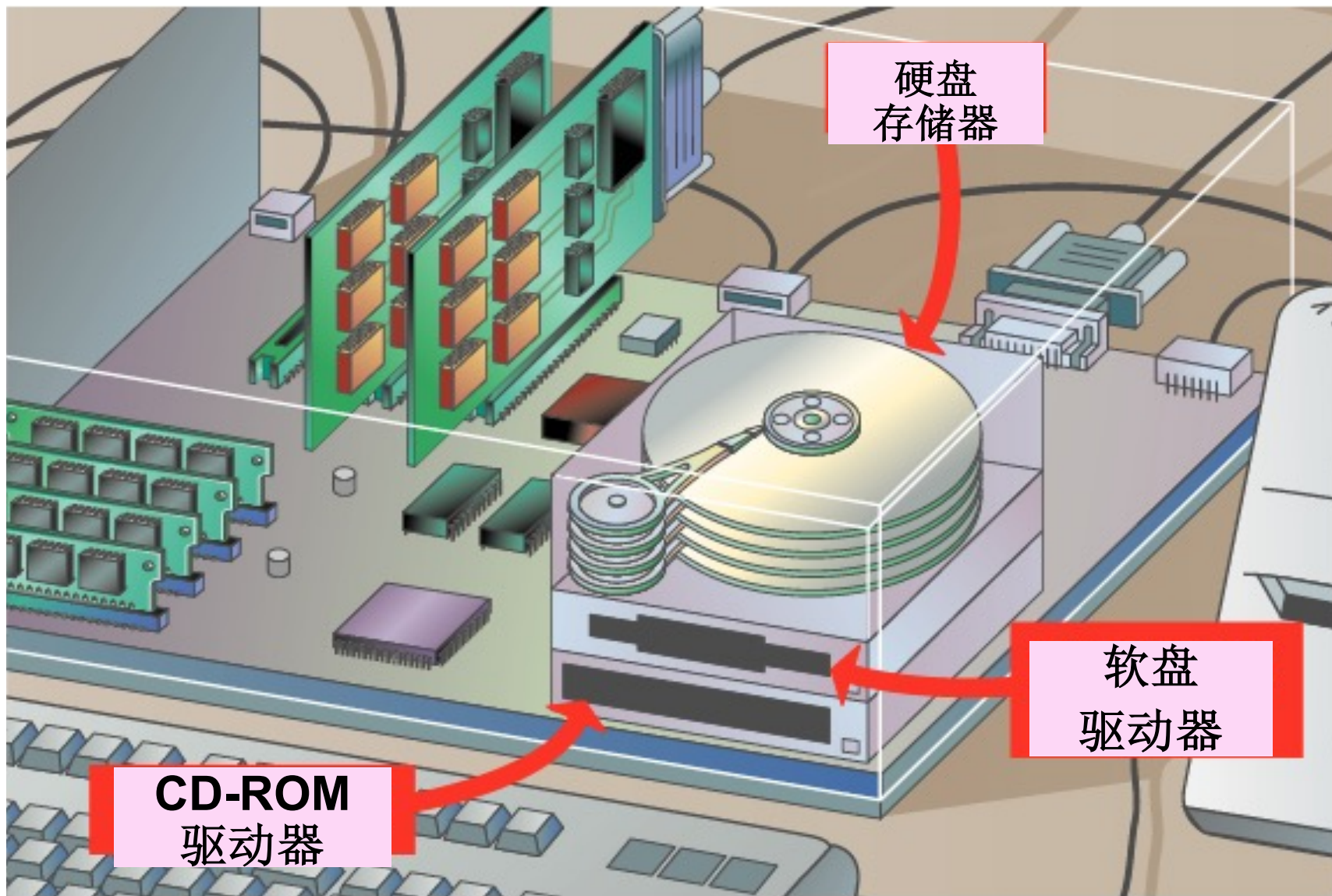
4.5 辅助存储器

一、概述

1. 特点 不直接与 CPU 交换信息



列出的时间和容量会随时间变化，但数量级相对关系不变。



三、硬磁盘存储器

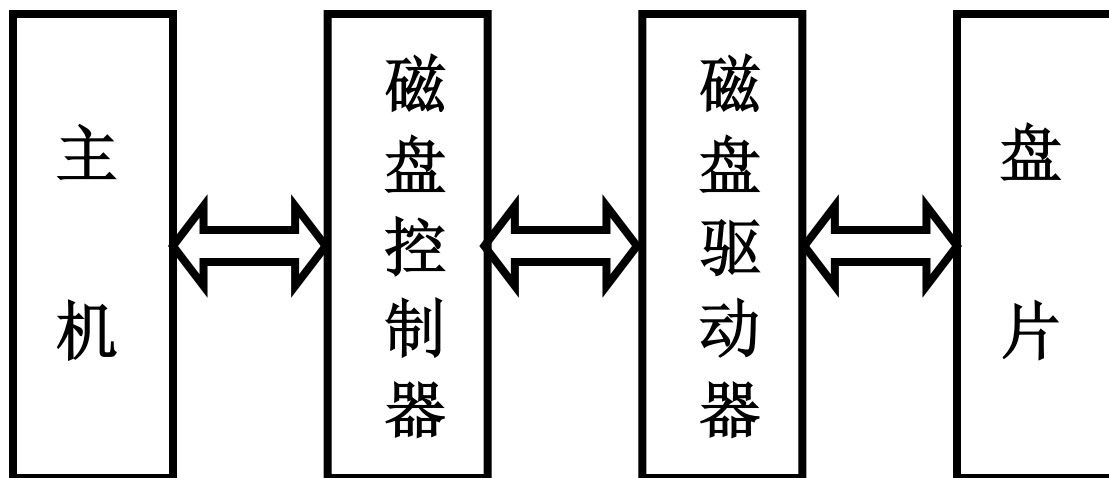
4.5

1. 硬磁盘存储器的类型

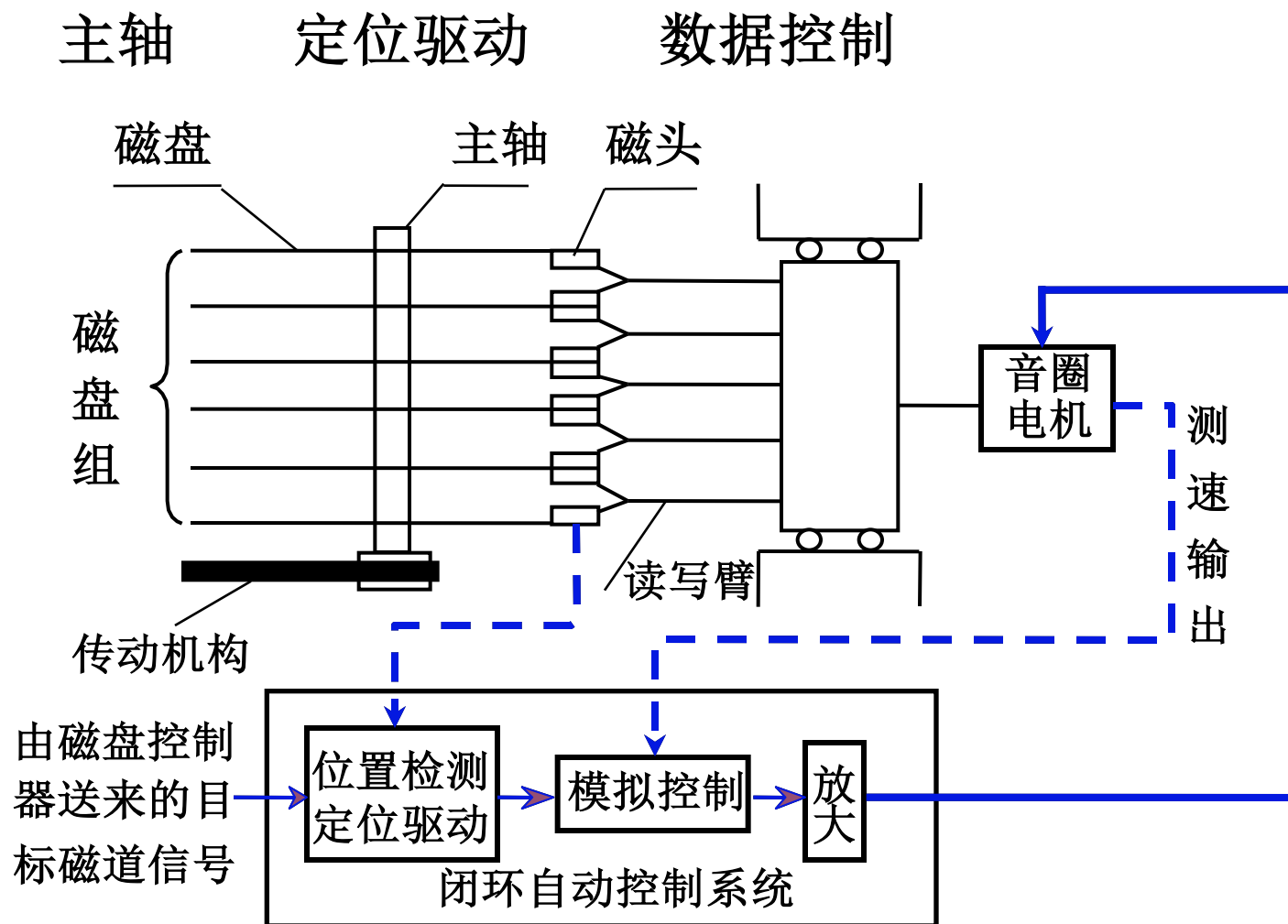
(1) 固定磁头和移动磁头

(2) 可换盘和固定盘

2. 硬磁盘存储器结构



(1) 磁盘驱动器



(2) 磁盘控制器

- 接收主机发来的命令，转换成磁盘驱动器的控制命令
- 实现主机和驱动器之间的数据格式转换
- 控制磁盘驱动器读写

磁盘控制器 是

主机与磁盘驱动器之间的 接口 { 对主机 通过总线
对硬盘 (设备)

(3) 盘片

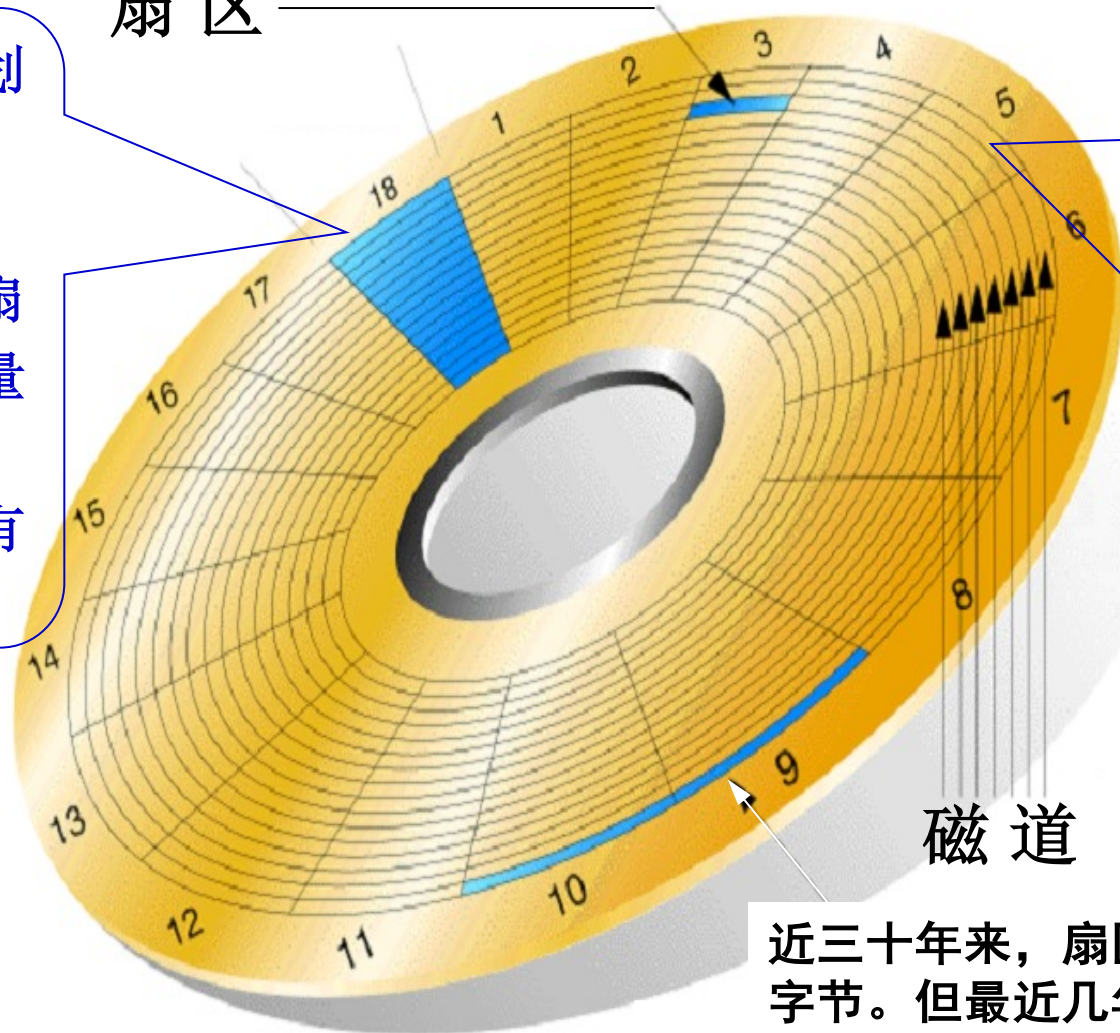
由硬质铝合金材料制成

磁盘的磁道和扇区

4.5

每个磁道被划分为若干段（段又叫扇区），每个扇区的存储容量为**512**字节。每个扇区都有一个编号

扇区



磁盘表面被分为许多同心圆，每个同心圆称为一个磁道。每个磁道都有一个编号，最外面的是**0**磁道

磁道

近三十年来，扇区大小一直是512字节。但最近几年正迁移到更大、更高效的4096字节扇区，通常称为4K扇区。国际硬盘设备与材料协会（IDEMA）将之称为高级格式化。

2. 磁表面存储器的技术指标

(1) 记录密度 道密度 D_t 位密度 D_b

(2) 存储容量 $C = n \times k \times s$

(3) 平均寻址时间 寻道时间 + 等待时间

辅存的速度 { 寻址时间
 磁头读写时间

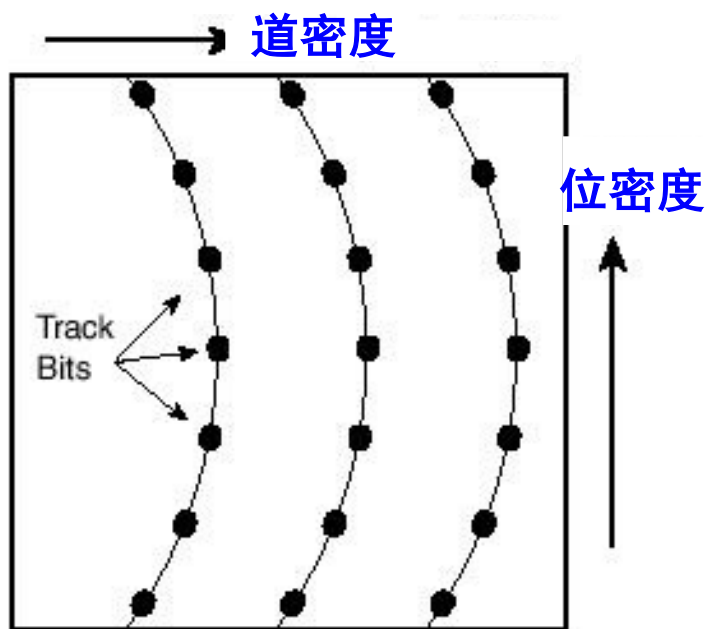
(4) 数据传输率 $D_r = D_b \times V$

(5) 误码率 出错信息位数与读出信息的总位数之比

如何增大磁盘片的容量？

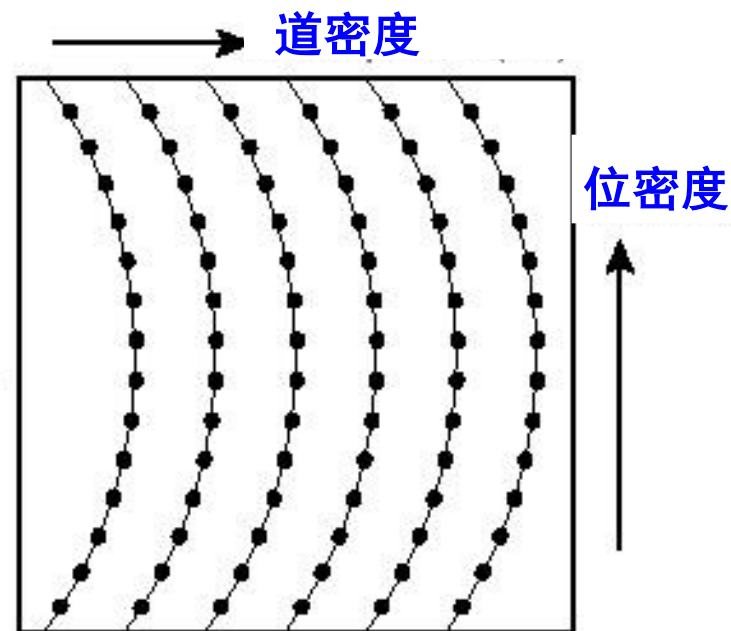
4.5

- 提高盘片上的信息记录密度！
 - 增加磁道数目——提高磁道密度
 - 增加扇区数目——提高位密度，并采用可变扇区数



低密度存储示意图

早期磁盘所有磁道上的扇区数相同，所以位数相同，内道上的位密度比外道位密度高

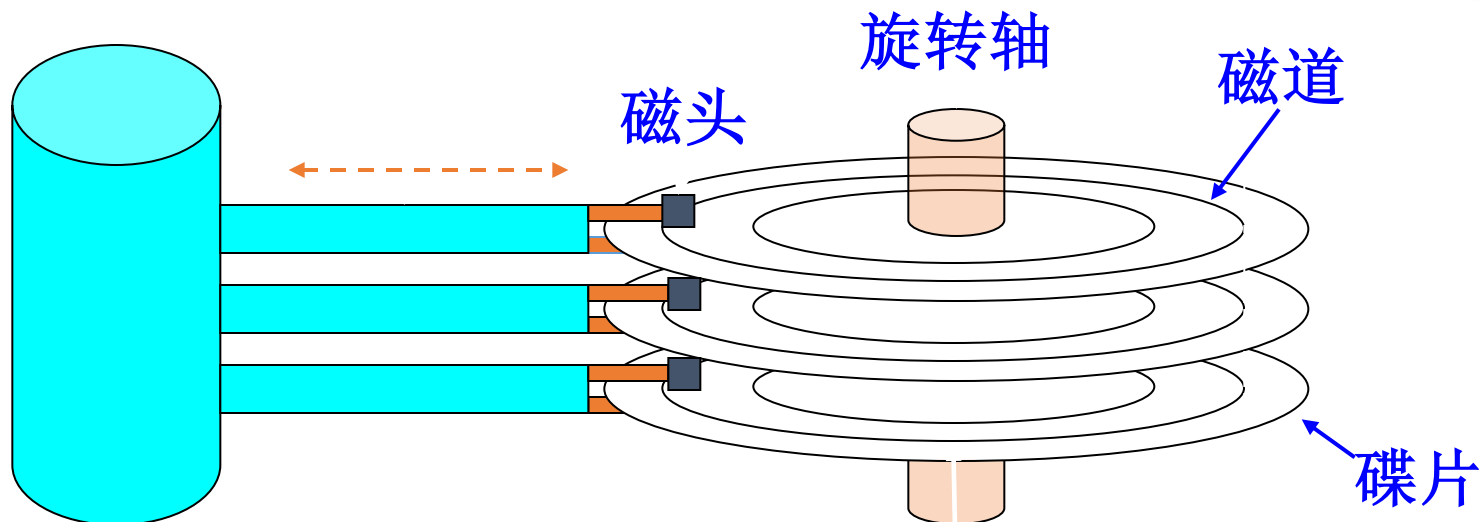


高密度存储示意图

现代磁盘磁道上的位密度相同，所以，外道上的扇区数比内道上扇区数多，使整个磁盘的容量提高

平均存取时间

4.5



硬盘的操作流程如下：

所有磁头同步寻道（由柱面号控制）→ 选择磁头（由磁头号控制）→
被选中磁头等待扇区到达磁头下方（由扇区号控制）→ 读写该扇区中数据

- 磁盘上的信息以扇区为单位进行读写，平均存取时间为：

$T = \text{平均寻道时间} + \text{平均旋转等待时间} + \text{数据传输时间（忽略不计）}$

- **平均寻道时间**——磁头寻找到指定磁道所需平均时间 (大约5ms)

- **平均旋转等待时间**——指定扇区旋转到磁头下方

磁盘响应时间计算举例

4.5

- 假定每个扇区512字节，磁盘转速为5400 RPM，声称寻道时间（最大寻道时间的一半）为12 ms, 数据传输率为4 MB/s, 磁盘控制器开销为1 ms, 不考虑排队时间，则磁盘响应时间为多少？

$$\begin{aligned}\text{Disk Response Time} &= \text{Seek time} + \text{Rotational Latency} + \text{Transfer time} \\ &\quad + \text{Controller Time} + \text{Queuing Delay} \\ &= 12 \text{ ms} + 0.5 / 5400 \text{ RPM} + 0.5 \text{ KB} / 4 \text{ MB/s} + 1 \text{ ms} + 0 \\ &= 12 \text{ ms} + 0.5 / 90 \text{ RPS} + 0.125 / 1024 \text{ s} + 1 \text{ ms} + 0 \\ &= 12 \text{ ms} + 5.5 \text{ ms} + 0.1 \text{ ms} + 1 \text{ ms} + 0 \text{ ms} \\ &= 18.6 \text{ ms}\end{aligned}$$

如果实际的寻道时间只有1/3的话，则总时间变为10.6ms，这样旋转等待时间就占了近50%！

$12/3 + 5.5 + 0.1 + 1 = 10.6 \text{ ms}$ 所以，磁盘转速非常重要！

为什么实际的寻道时间可能只有1/3？

访问局部性使得每次磁盘访问大多在局部几个磁道，实际寻道时间变少！