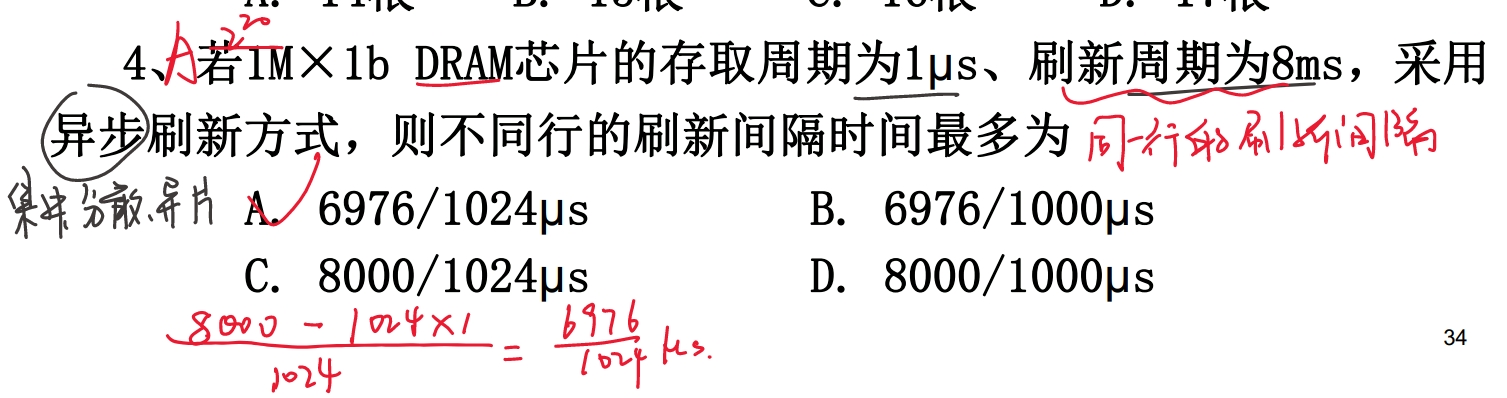
1. **存储器**

**一**．存储器概述（存储器的层次结构）

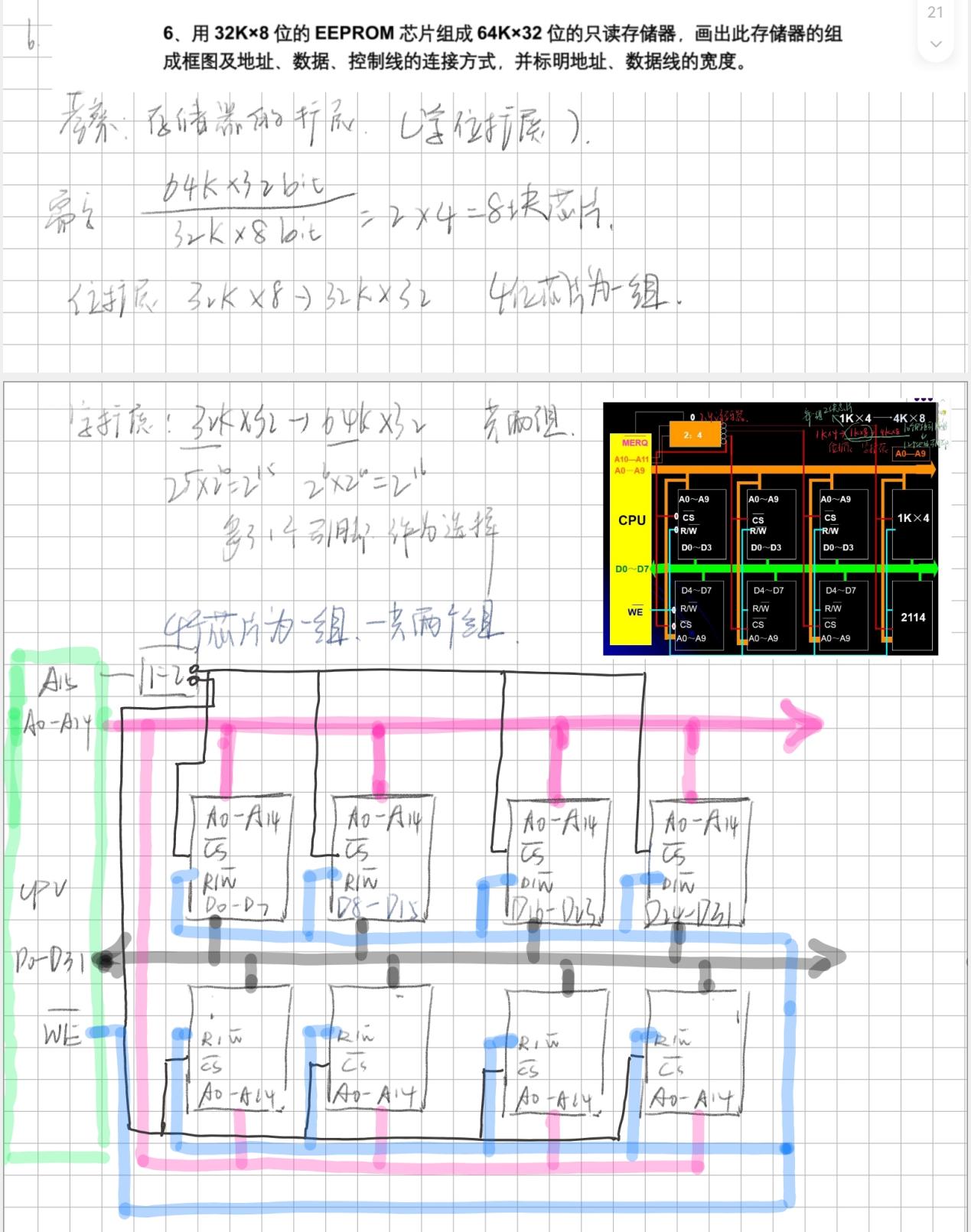
1. 为什么需要多种存储器（缓存cache，内存，硬盘）：实现速度、容量和价格的折中
2. 如何实现速度、容量和价格的折中：将更频繁使用的数据放进靠近CPU的存储器中
3. 各种存储器的容量如何确定：价格越高的存储器，容量越小
4. 内存越大，计算机运行速度越快；而硬盘越大，计算机存储容量越大。为什么：内存存储的是最频繁使用的数据，而外存（辅存）主要用于存储计算机中的数据。
5. 数据的传输过程是逐级搬运的
6. 构成上看：内存=寄存器+Cache+主存，存储容量上看：内存=主存
7. 存储器带宽：单位时间里存储器所存取的信息量，bit/s byte/s
8. 相联存储器是按内容指定方式进行寻址的存储器
9. 解决成本和性能之间的矛盾的策略--存储器分层
10. 构成多级存储系统的依据就是程序的局部性原理
11. **程序的局部性原理**：这种在某一段时间内频繁访问某一局部存储器地址空间，而对此范围以外的地址空间则很少访问的现象成为程序的局部性原理。
12. 时间局部性：最近被访问的信息很可能还要被访问
13. 空间局部性：最近被访问的信息临近地址的信息也可能被访问。
14. 三级存储系统
15. 内存储器：正在执行的程序的指令和数据存储在**CPU能直接访问**的存储器中。常规的内存被称为主存。
16. 辅存：为了扩大存储器的容量，在内存储器之外增加容量更大但访问速度稍慢的外存储器（外存），或称之为辅助存储器（辅存）。**CPU是无法直接访问辅存的**。
17. 高速缓冲存储器Cache
18. SRAM（static）--存取**速度快**，是Cache的存储单元
19. SRAM特点：（存储位元是一个触发器，具有两个稳定的状态）
20. 读取时不会因为读操作导致数据发生变化；
21. 随着时间的流逝，SRAM中的数据不会遗失；
22. SRAM存储容量的计算
23. SRAM采用双译码方式（减少选择线的数目）
24. 引脚数目：地址线+数据线+控制线（三条）
25. DRAM（动态）--存储**容量大**，是主存的存储单元
26. DRAM的特点：（存储位元是由一个MOS晶体管和电容器组成）
27. 读取时会因为读操作导致数据发生变化（原因：由电容构成）
28. 随着时间的流逝，SRAM中的数据会遗失（原因：电容放电）
29. DRAM存储容量的计算
30. DRAM芯片的逻辑结构分析：
31. 采用双译码方式，但地址引脚是地址空间的一半（有CAS RAS列/行选地址信号线）
32. 片选信号CS没有了（用CAS RAS进行代替）
33. 刷新控制与定时：设定固定时间，记录我什么时候该刷新
34. 刷新计数器：记录当前刷新到哪一行
35. 刷新操作：是一行一行进行刷新，每次刷新把该行每列都进行刷新操作
36. 芯片的刷新方式：

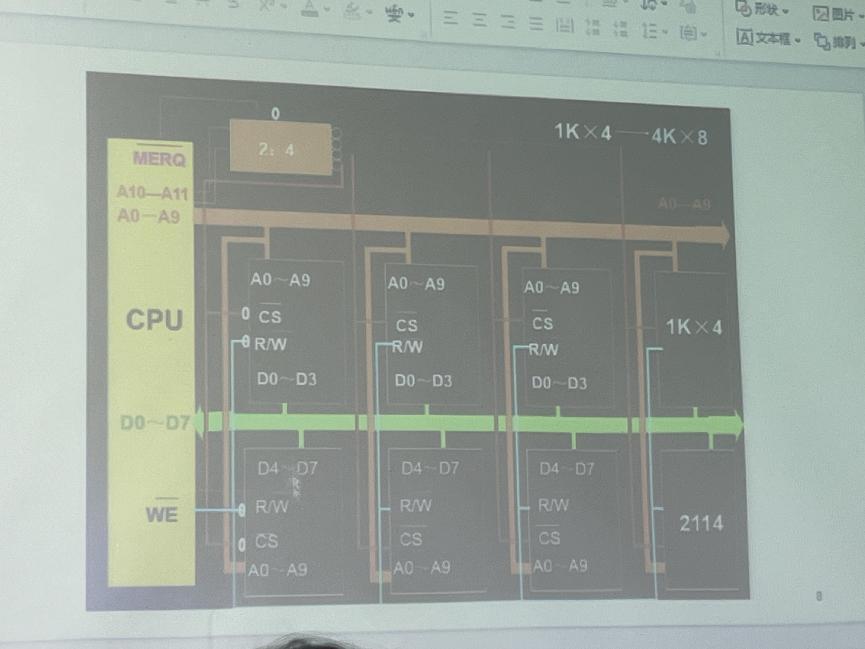
**刷新周期：指同一存储元刷新的最大间隔时间（同一行的刷新间隔）**

1. 集中式刷新：各行刷新连续进行，存在死区（此时CPU无法访存）
2. 分散式刷新：各个刷新分散在每个T（指读取周期）中，无死区，但T会翻倍
3. 异步式刷新：各行刷新均匀分布在刷新周期中，无死区，T不变



1. 引脚数目：地址线（减半）+数据线+控制线（三条）
2. SRAM和DRAM的区别
3. DRAM中增加了行地址锁存器和列地址锁存器。由于DRAM容量很大，地址线的数目相当多，为了减少芯片引脚的数量，将地址分为行，列两部分分时传送。此外，片选信号CS也由新增的RAS,CAS信号实现。
4. SRAM采用半导体管的“导通”和“截止”来记忆，只要不掉电，存储信息就不会丢失。DRAM的信息是用电荷存储在电容上的，随着时间的推移，电荷会逐渐漏掉，存储信息也会丢失，因此需要周期性的对其“刷新”。
5. 存储器容量扩充
6. 位扩展（扩展数据引脚）：地址线和控制线公用而数据线单独分开连接
7. 字扩展（存储单元数目增加）：地址总线和数据总线公用，控制总线中的R/W公用，但片选使能端CS不能公用
8. 字位扩展：位扩展（地址线、控制线公用，数据线分开）+字扩展（地址线，数据线公用，控制线R/W公用但CS分开，CS接一个译码器）会画图
9. 考题注意：看是两个相同类型的存储器进行组合还是不同类型的存储器进行组合





1. 高级的DRAM结构--原理：程序的局部性原理
2. FPM DRAM（fast page mode）

（1）发出一个行信号，输出同一行的多列数据，节省了输出多个行信号的时间

1. CDRAM（cache）
2. DRAM内部集成了一个小容量的SRAM
3. 用DRAM存，从SRAM中拿
4. FPM DRAM和CDRAM区别：

①FPM是拿出之后直接输出

②CDRAM是拿出之后先放进SRAM里，要用的时候再从SRAM里面去拿

1. SDRAM（synchronous同步）
2. 优化原理1：采用同步工作方式，操作者中途可以离开（做其他工作）
3. 优化原理2：支持突发传输模式（多个数据/操作），提高MEM带宽
4. FPM DRAM和SDRAM区别

①FPG是发出行信号后，发出一系列列信号

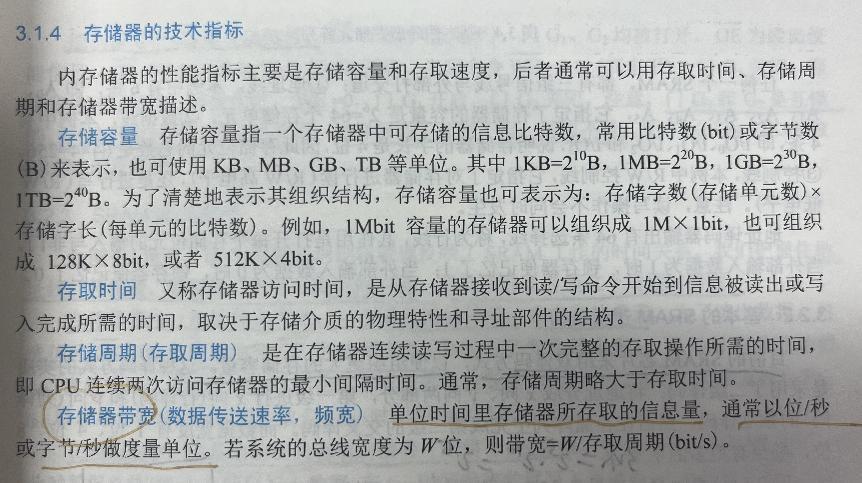
②SDRAM发出一个信号后，不再发出其他信号，而是CPU猜测会需要哪些数据（局部性原理）

1. 只读存储器ROM
2. 分类：
3. 掩膜ROM：存储内容固定的ROM，由生产产家提供产品
4. 可编程ROM：用户后写入内容，有些还可以多次写入

①一次性编程：PROM（可一次性写入的）

②多次编程：EPROM（光擦除）,EEPROM（电擦除）

1. 存储器的性能指标 书65



存储器性能指标的考题见八中的经典例题

1. 并行存储器
2. 双端口：所有引脚数量加倍，2个地址口，2个数据口。对不同的地址单元同时进行读/写操作）
3. 多模块交叉存储器：
4. 存储器的模块化组织：顺序、交叉

①顺序：特点：某模块进行存取时，其他模块不工作。优点是鲁棒性强，某一模块出现故障，其他模块照常工作；通过增添模块来扩充存储器容量比较方便。缺点是各模块串行工作，存储器的带宽受限。

②交叉：连续地址分布在相邻的不同模块内，同一个模块内的地址是不连续的。优点是对连续字的成块传送可实现多模块流水式并行存取，大大提高存储器的带宽，适合批量数据读取。缺点：鲁棒性差。

1. 多模块交叉存储器结构与CDRAM相似，但区别在于：

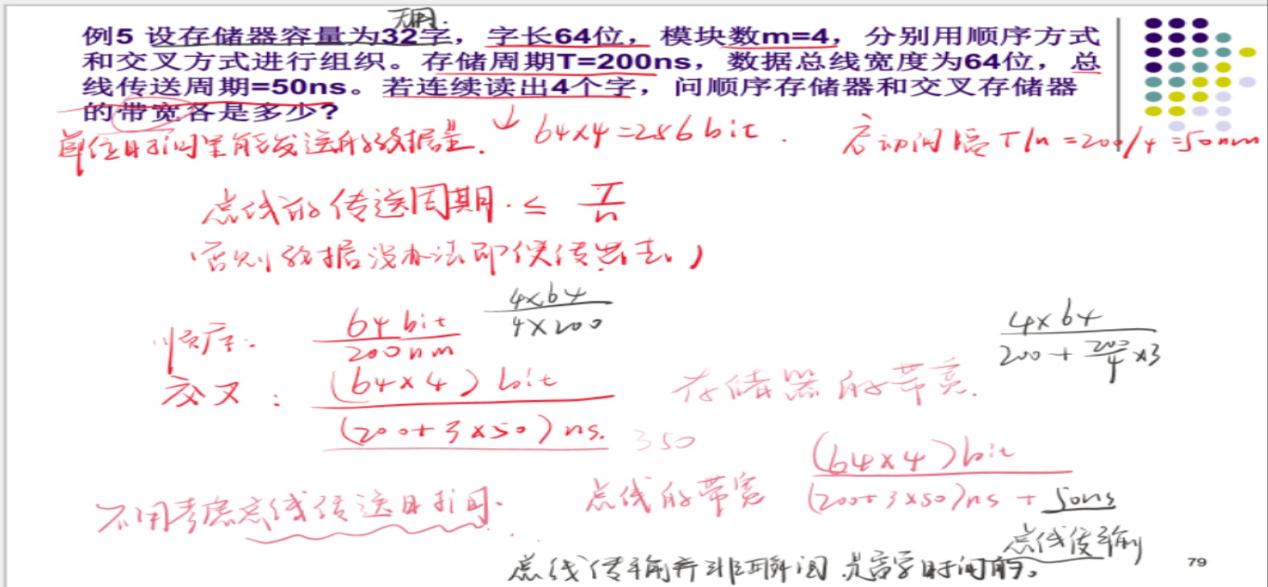
①CDRAM每次只能访问同一行

②此处RAS接的不同的信号，可以访问不同的行

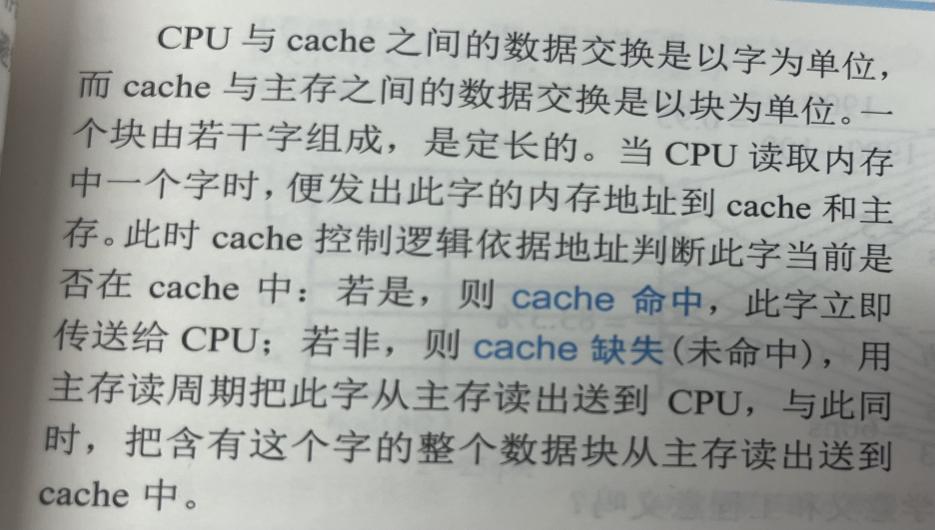
1. 时间：

①顺序：t = x\*T；②交叉：t=T+(X-1)\*(T/n)（这里的n是模块数）

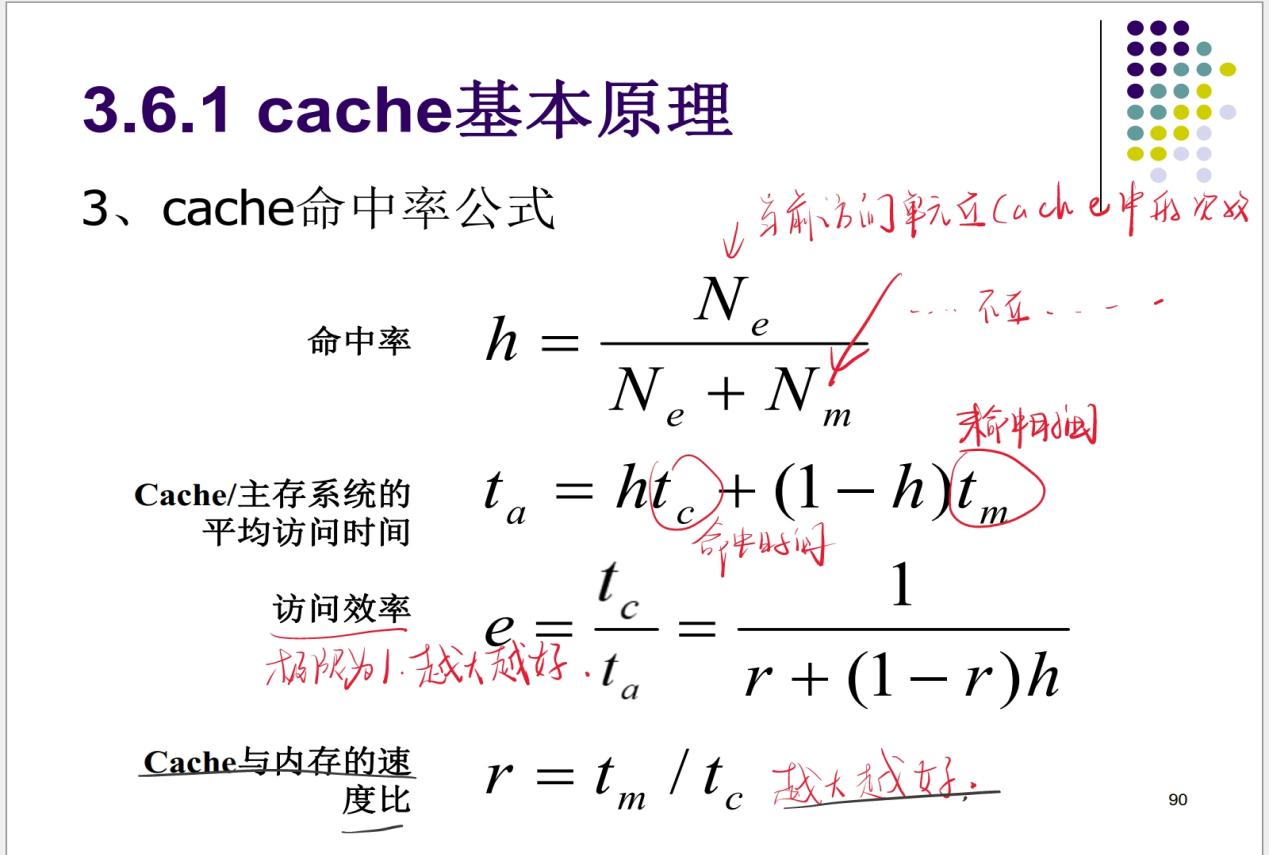
1. 经典例题：

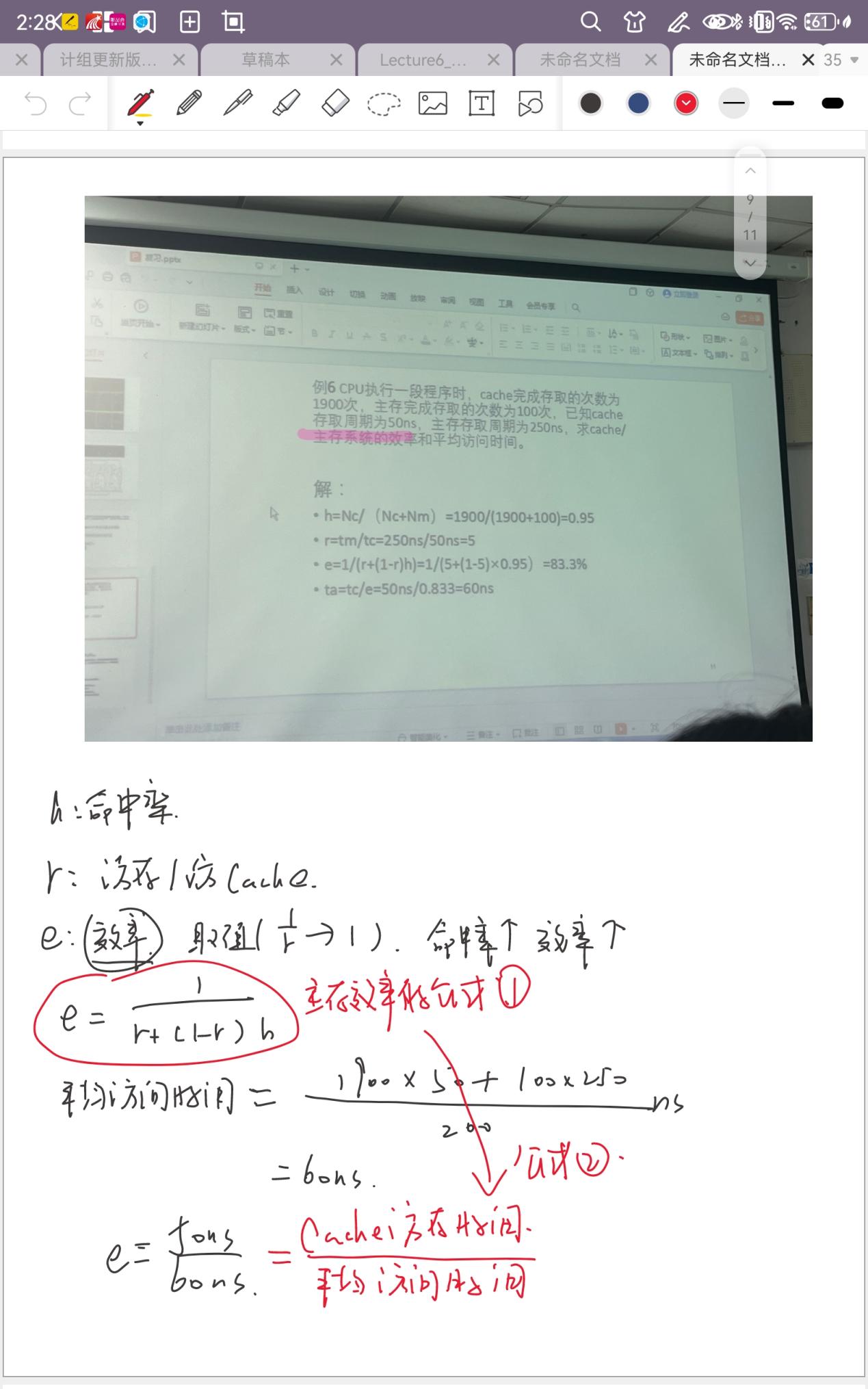


1. Cache
2. Cache的基本原理：
3. 程序运行时，CPU使用的一部分数据/指令会预先成批拷贝在Cache中，Cache的内容是主存中部分内容的映像。
4. CPU访问Cache时，经过主存块地址和Cache中的标志进行比较，判断出访问的内容在Cache中，这种情况称之为命中。
5. Cache的工作原理

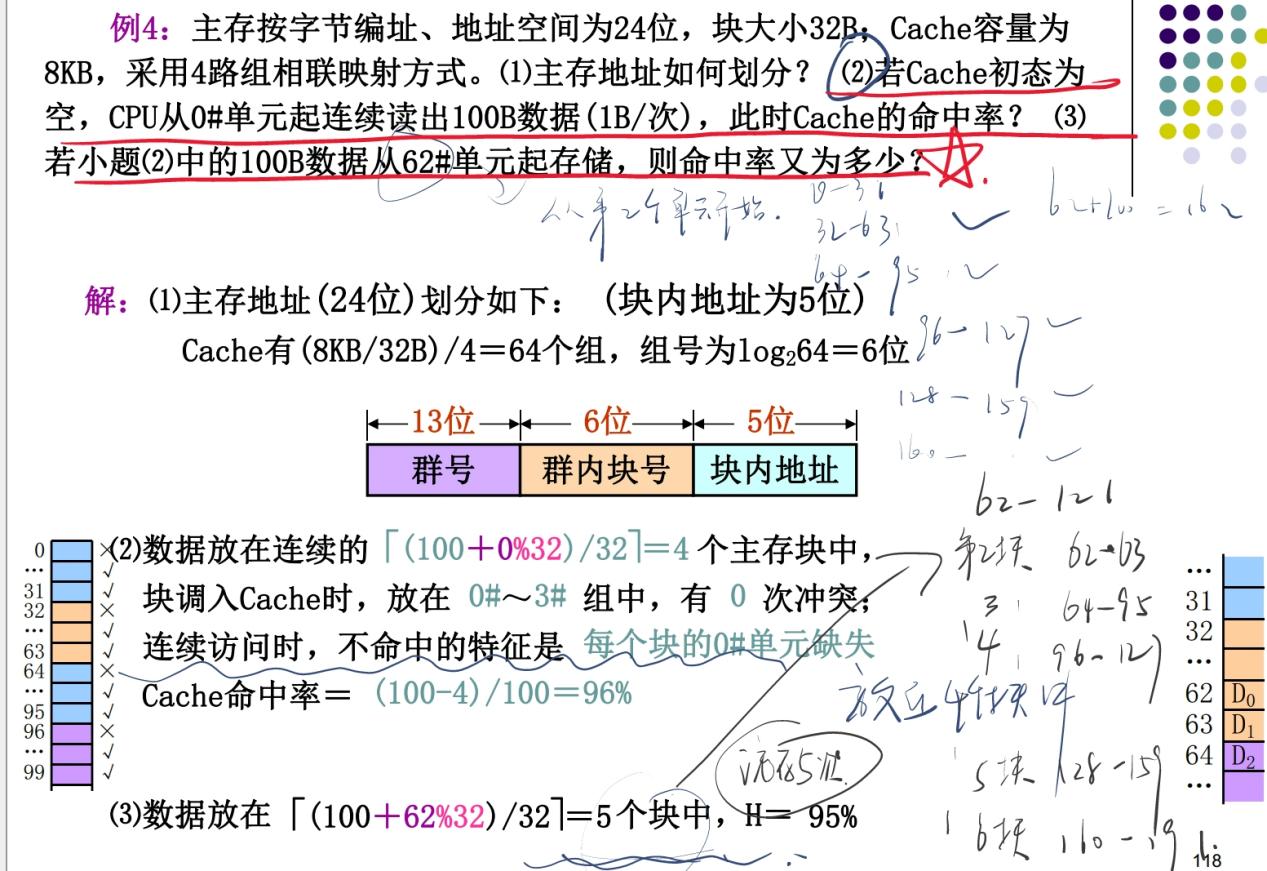


1. Cache的性能指标 书93
2. 命中率
3. 平均访问时间
4. 访问效率（也即主存系统的效率） = Cache的访存时间 / 平均访存时间
5. Cache和内存的速度比





例题：



1. 主存和cache的地址映射
2. 全相连的映射方式：**主存块i可放到cache的任意第j行中**

①地址划分：主存地址：

|  |  |
| --- | --- |
| 主存块号i | 块内地址w |

cache地址：

|  |  |
| --- | --- |
| 行号j | 块内地址w |

②标记选定：

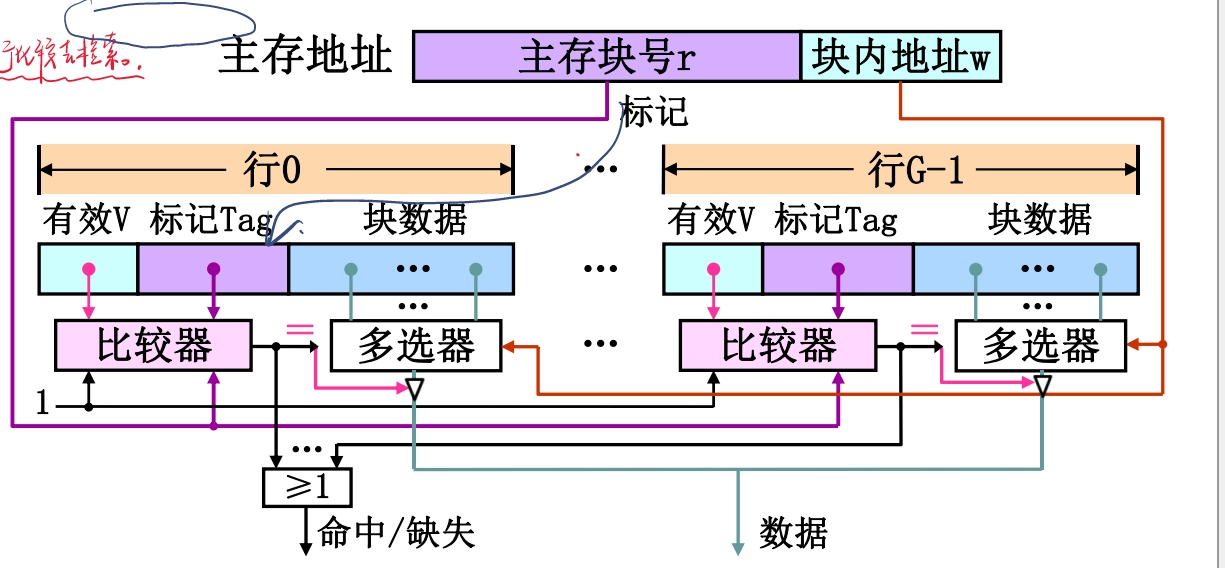
索引：无

标记=主存块号

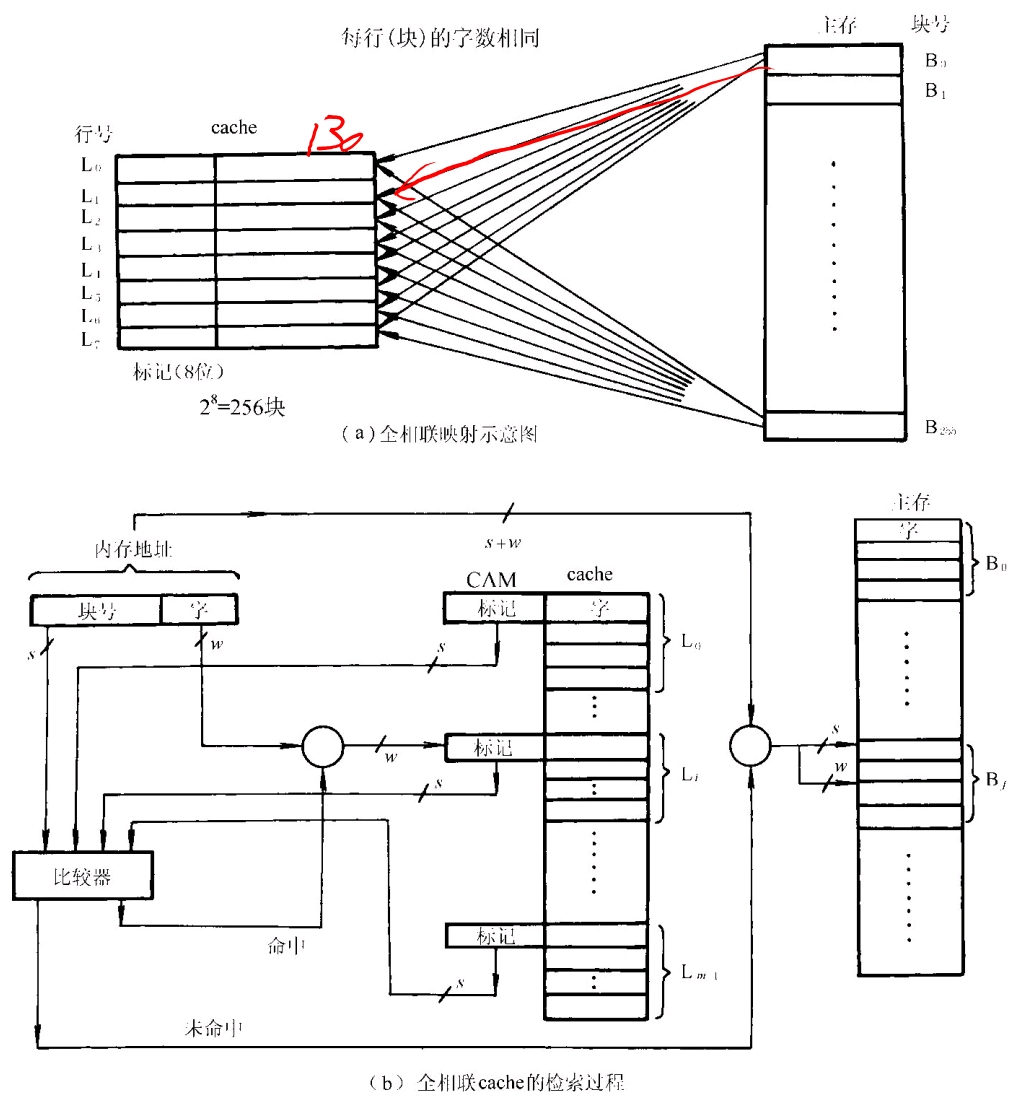
③特征：块调入时**冲突率最低**，地址变换速度较快，成本最高（比较器数目多）

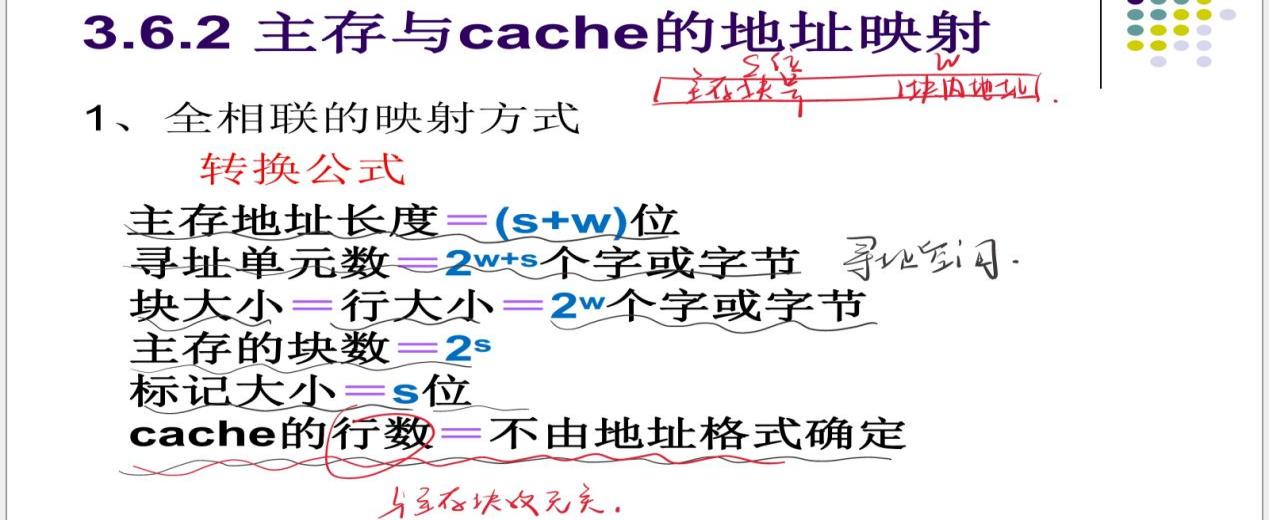
④优点：冲突概率小，Cache利用率高；缺点：比较器难实现；应用：小容量的cache

⑤为什么需要比较所有的块？--全相联映射中主存块号和Cache中的行号之间没有任何联系，所以无法根据索引直接找到存储的位置，需要在Cache中逐行比较去检索。

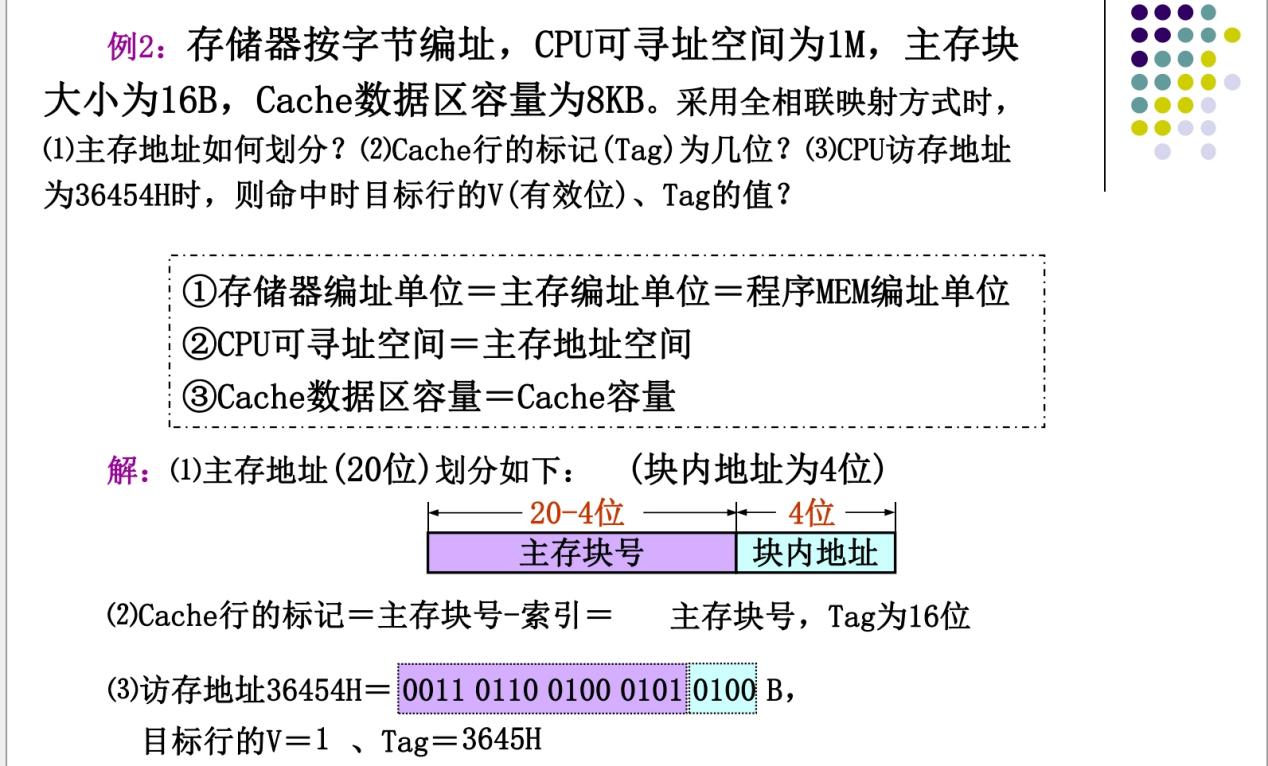


⑥检索过程：CPU的访存指令指定了一个主存地址，为了快速检索，**指令中的块号与Cache中所有行的标记同时**在比较器中进行比较（只有一个比较器，每行分时比较）。若有多个比较器，则同时多行比较。不管哪种方式比较器的成本都很高。





例题：



1. 直接映射方式：**主存块i可放到cache的某个行j中**

公式：i = j mod m（m位Cache中的行数）

①地址划分：主存地址：

|  |  |  |
| --- | --- | --- |
| 区号r | 区内快号j | 块内地址w |

cache地址：

|  |  |
| --- | --- |
| 行号j | 块内地址w |

②标记选定：

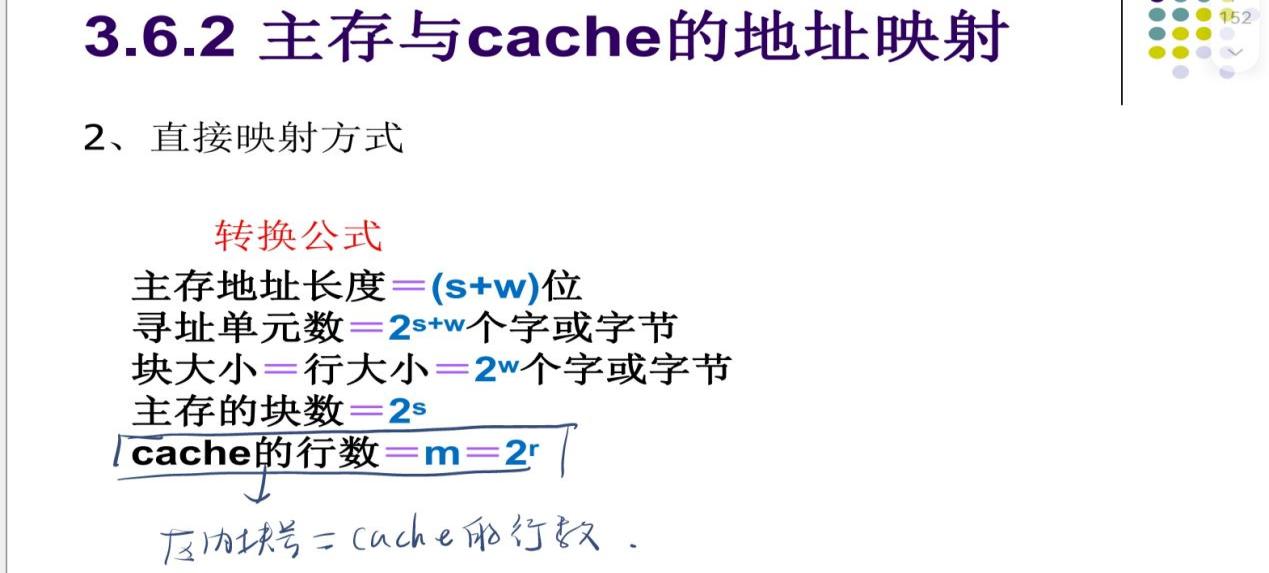
索引：区内块号

标记=主存块号-区内块号=区号

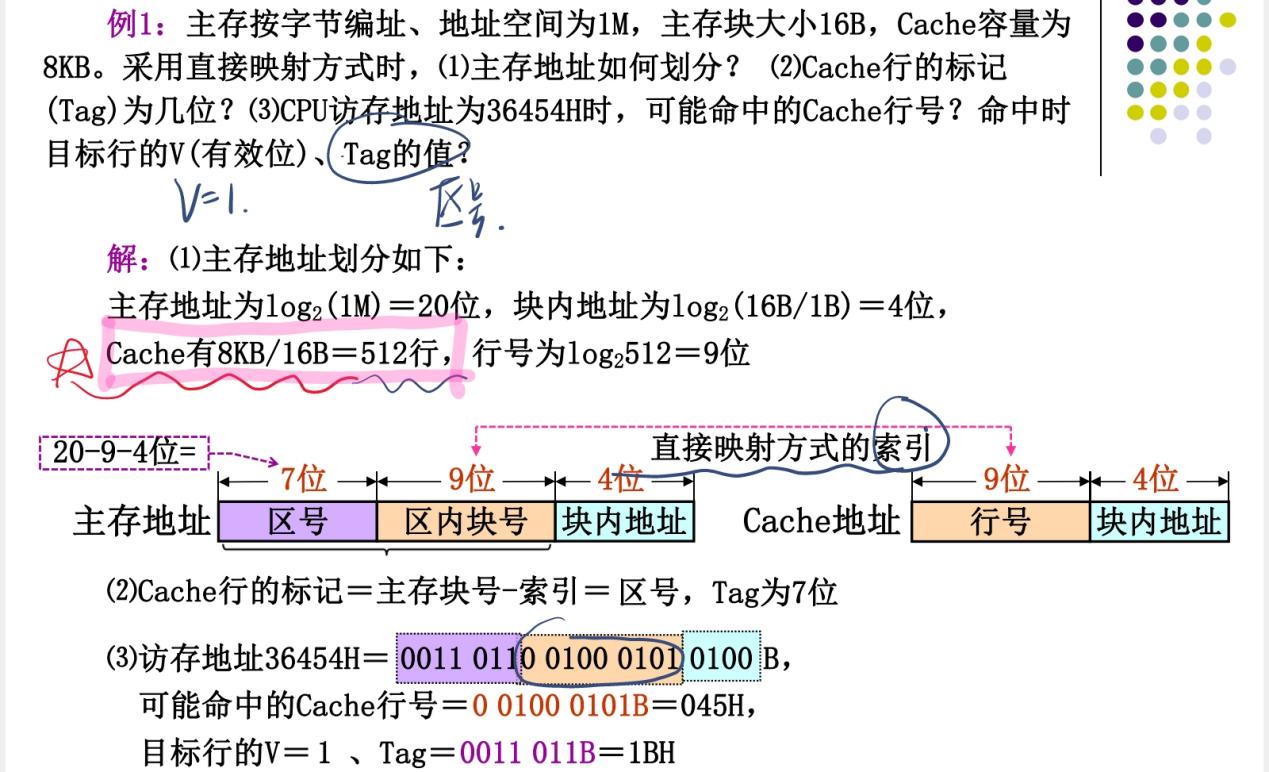
③特征：块调入时冲突率最高，地址变换速度最快（低几位），成本最低

④优点：比较电路少，硬件实现简单，cache地址即为主存地址低几位，不需变换；缺点：冲突率高；应用：大容量的cache

⑤检索过程：CPU的访存指令指定了一个主存地址，**先用区内块号快速索引到Cache中对应的行处，再将Cache行中的标记位和和区号进行比较**。



例题：



1. 组相联映射方式：**将cache进行分组，每个组有k个行（k路组相联）；主存块i可放到第j组的任意行中**

①地址划分：主存地址：

|  |  |  |
| --- | --- | --- |
| 群号 | 群内块号d | 块内地址w |

cache地址：

|  |  |  |
| --- | --- | --- |
| 组号d | 组内行号s | 块内地址w |

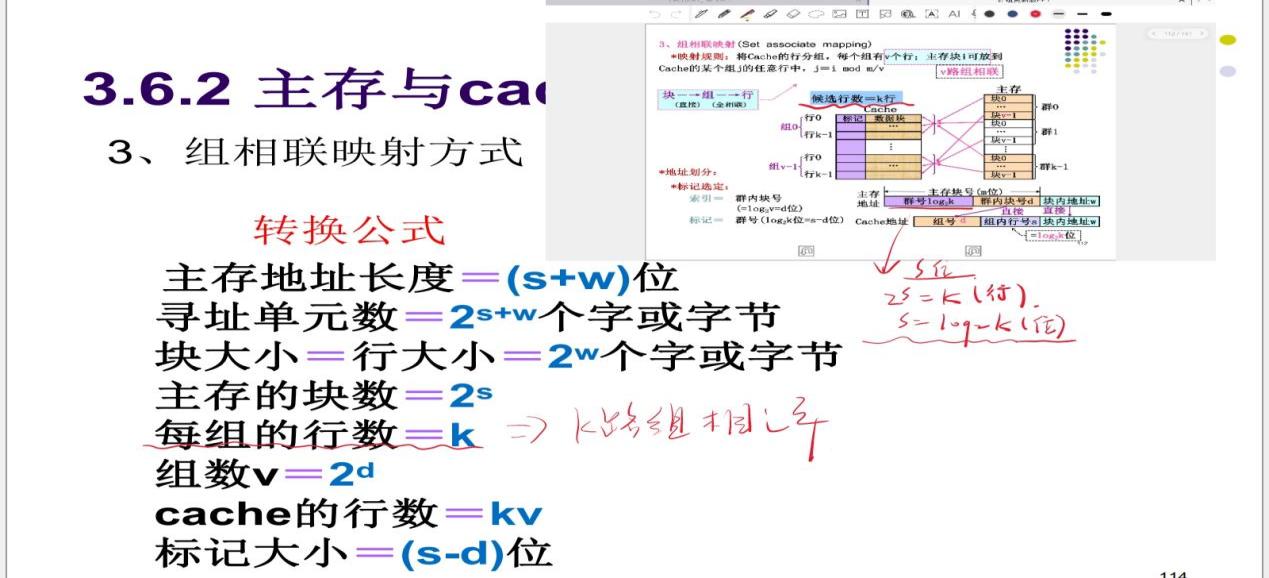
②标记选定：

索引：群内块号

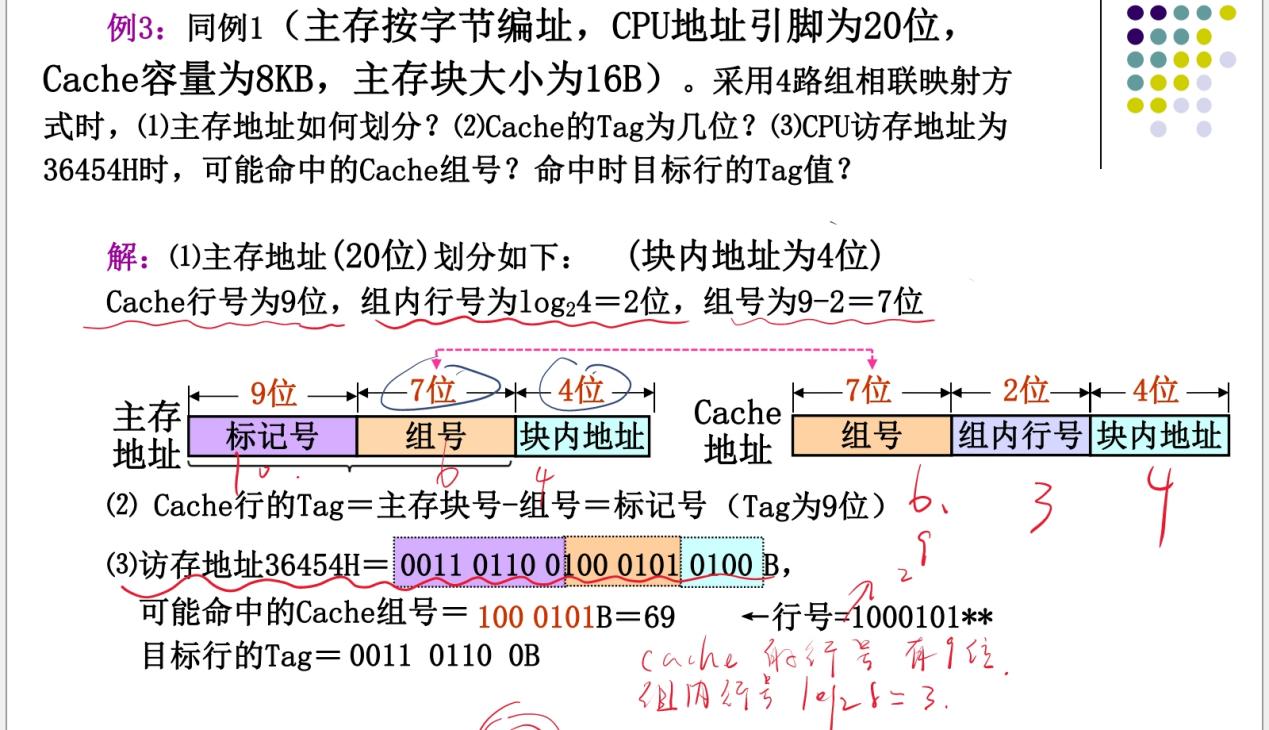
标记=主存块号-群内块号=群号

③特征：块调入时冲突率较低，地址变换速度较快，成本较低

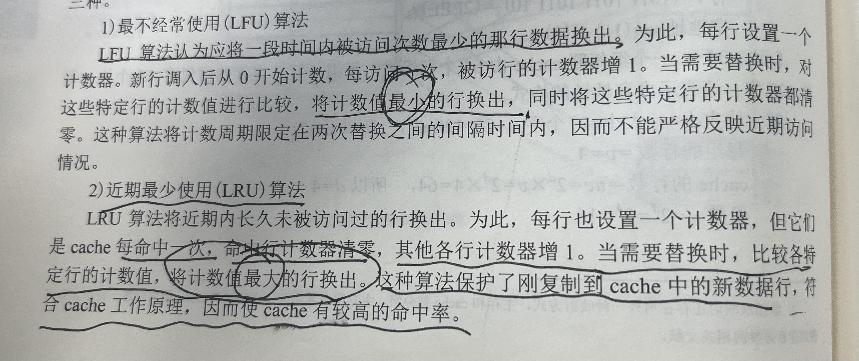
④检索过程：CPU的访存指令指定了一个主存地址，**先利用群内块号快速索引到对应的Cache组中，再在该组中将每行的标记位和群号进行比较**。



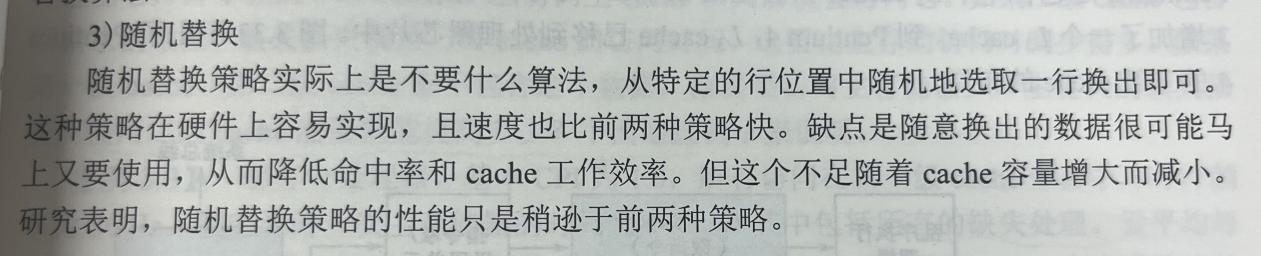
例题：



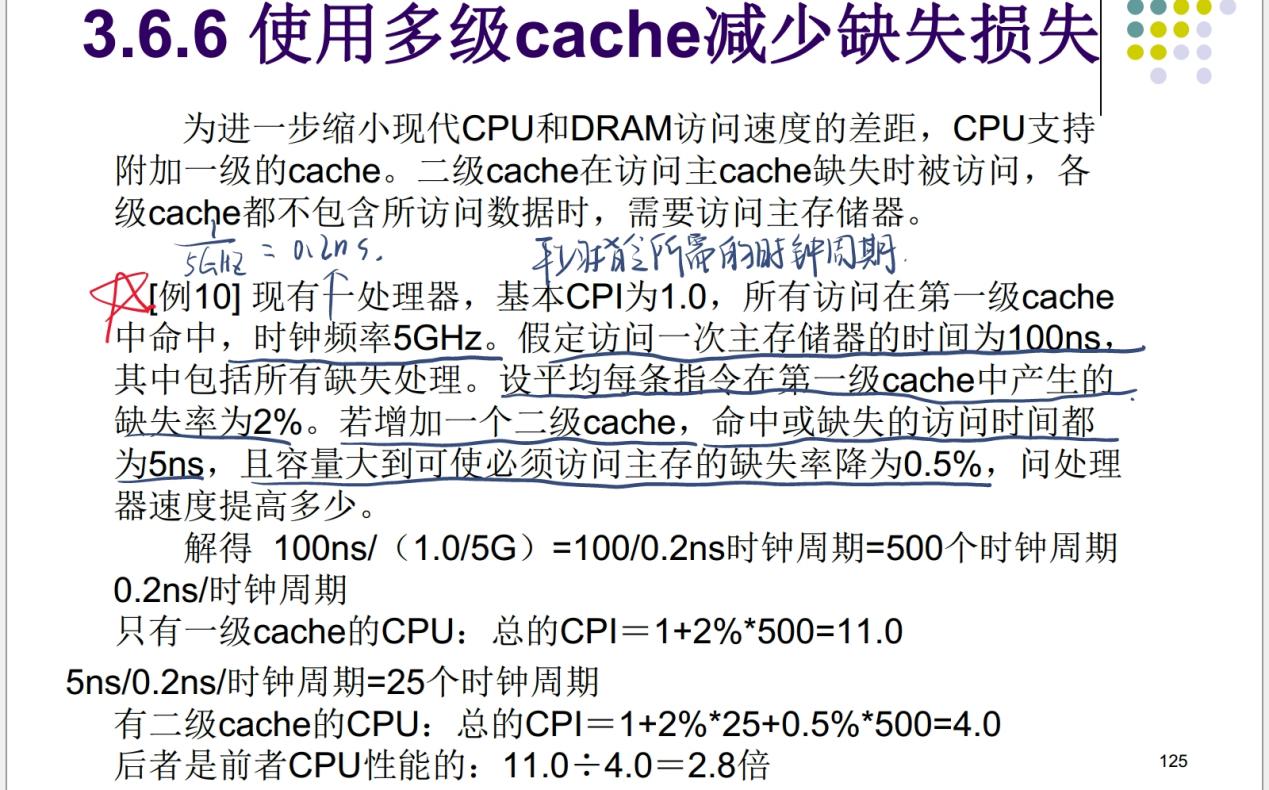
1. 替换策略：LFU（最不经常），LRU（近期最少），随机替换PPT 122页，书100
2. LFU：将计数值最小的换出
3. LRU：将计数值最大的换出，这种算法保护了刚复制到Cache中的新数据行，符合Cache的工作原理，因为使Cache有较高的命中率。



1. 随机替换策略不需要什么算法，从特定的位置随机选取一行换出



1. 写操作策略：（为保证写一致性）
2. **写回法**：只修改cache不修改主存，直到换出时
3. **全写法**：写命中时，cache和主存一起写
4. 写一次法：与写回法一致，但第一次cache命中时采用全写法
5. 使用多级cache减少缺失损失



1. 辅存：

（1）内存储器：正在执行的程序的指令和数据存储在**CPU能直接访问**的存储器中。常规的内存被称为主存。

（2）辅存：为了扩大存储器的容量，在内存储器之外增加容量更大但访问速度稍慢的外存储器（外存），或称之为辅助存储器（辅存）。**CPU是无法直接访问辅存的**。

1. 虚拟存储器 书104-105（仔细读）
2. 程序进行虚地址到实地址的转换过程称为程序的再定位。
3. 虚存的访问过程：每次访存时，首先判断该虚地址所对应的部分是否在实存中；如果是，则进行地址转换并用实地址访问主存；否则按照某种算法将辅存中的部分程序调度进内存，再进行地址转换并用实地址访问主存。
4. 每个程序的虚地址空间可以>>（扩容）/<<（地址变换，多用户多任务中：实存空间较大，而单个任务并不需要很大的地址空间，较小的虚存空间可以**缩短指令中地址字段的长度**）实地址空间
5. 有了虚存机制后，应用程序可以透明的使用整个虚存空间。对于应用程序，如果主存的命中率很高，虚存的访问时间接近主存的访问时间，而虚存的大小仅仅依赖于辅存的大小。
6. 虚拟存储器：具有辅存的大容量和接近主存的访问速度；概念模型，实际不存在。
7. **用户编址程序时使用的地址称为虚地址或逻辑地址，其对应的存储空间称为虚存空间或逻辑地址空间；而计算机物理内存的访问地址则称为实地址或物理地址，其对应的存储空间称为物理存储空间或主存空间。**
8. **计算机指令指定的访问主存储器所用的地址叫做逻辑地址。**
9. 根据主存-辅存间**信息传输单位的不同，虚拟存储器可分为三类：页式，段式，段页式**
10. cache和虚存的异同：书105

①出发点相同：构造性能接近高速的存储器，而价格和容量接近低速的存储器（cache-主存，主存-辅存）

②原理相同：程序的局部性原理

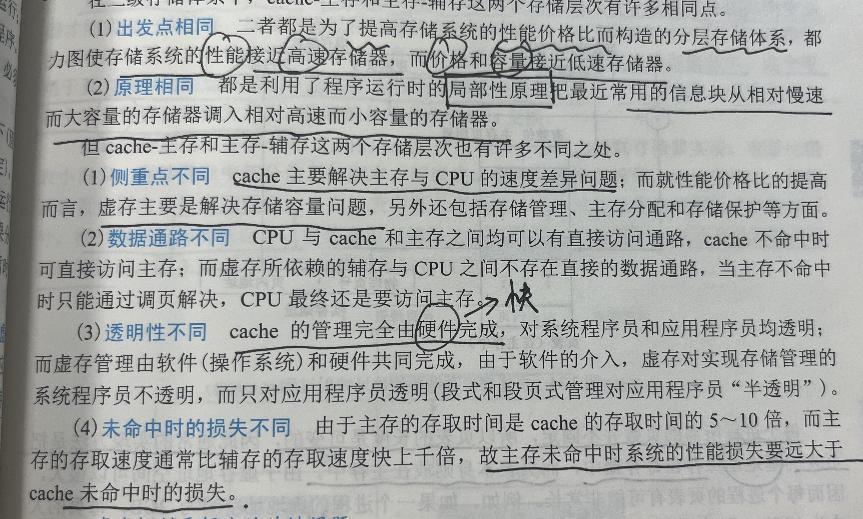
③侧重点不同：cache：速度差异；虚存：存储容量

④透明性不同：cache：纯硬件；虚存：软硬结合

⑤未命中时的损失：虚存>>cache

⑥数据通路不同：

注意：调页操作指的是将数据从硬盘调入内存，或从内存写回硬盘的过程。



虚拟存储器的地址管理方式：页式，段式，段页式

1. 页式虚拟存储器：
2. 页表：

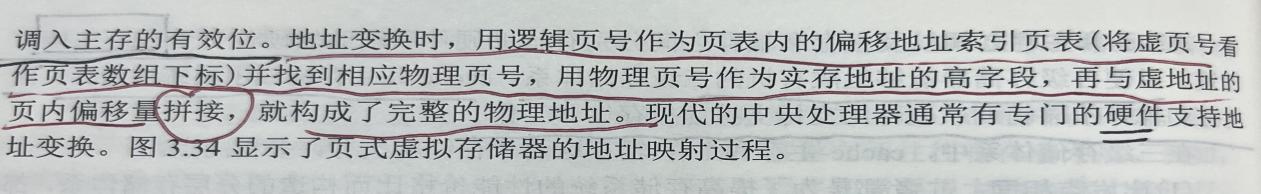
①当一个进程进行时，其页表中一部分放在主存中，另一部分放在辅存中保存

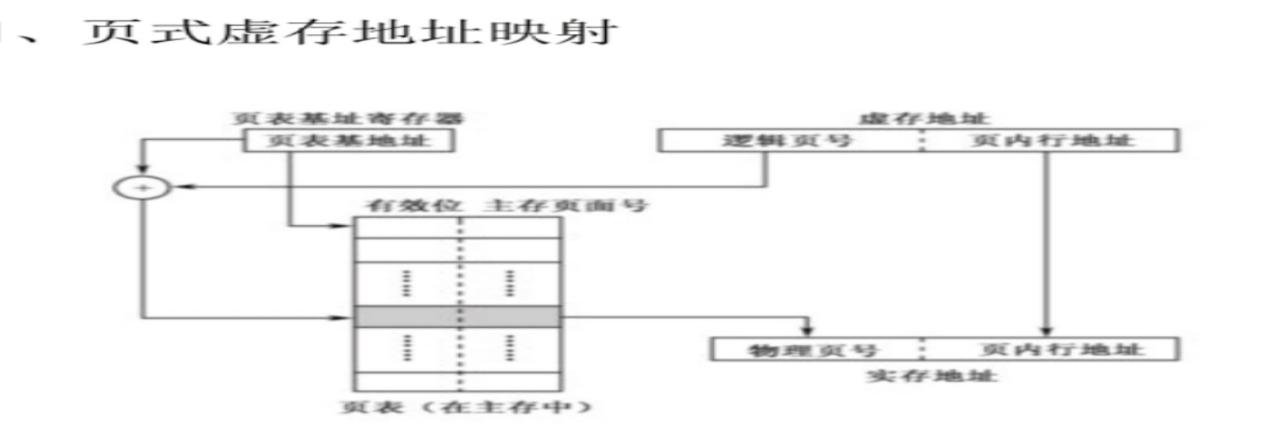
②内页表：虚地址到主存物理地址；外页表：虚地址到辅存物理地址（外页表通常放在辅存中，在需要时可调入主存中。当主存不命中时，有存储管理部件相CPU发出“缺页中断”，进行**调页**操作。

1. 转换后援缓冲器TLB

①把页表最活跃的部分放在高速存储器（Cache）中--TLB，又称快表；而保存在主存中的完整的页表则称为慢表（程序的局部性原理）

1. 特点：优点：页长固定，因而便于构造页表，易于管理，且不存在外碎片；缺点：页长和程序的逻辑大小无关
2. 地址映射过程：



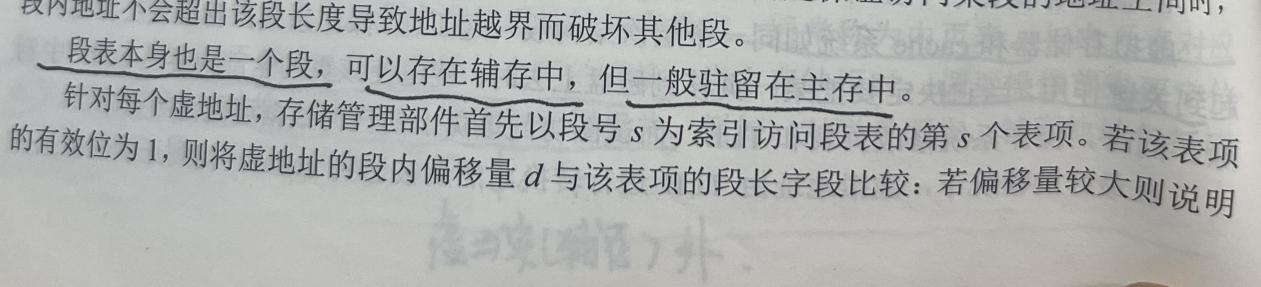


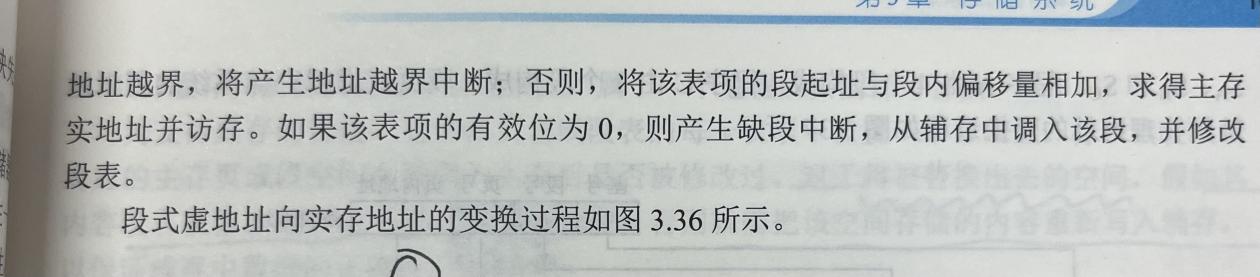
1. 段式虚拟存储器：
2. 段表：

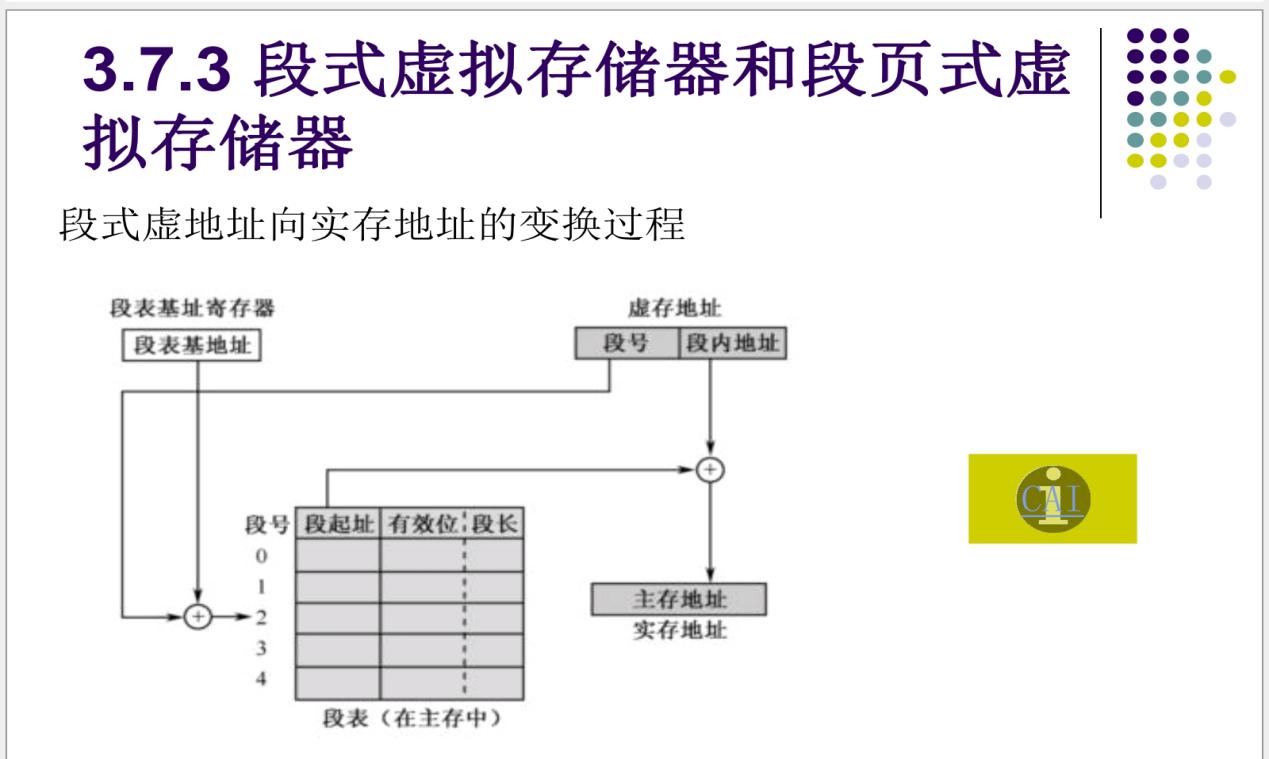
①有效位，段起址，段长

②段表本身也是一个段，可以在辅存中，但一般在主存中

1. 特点：优点：段的逻辑独立性使其易于编译、管理、修改和保护，也便于多道程序共享；段长可以根据需要动态改变，允许自由调度，以便有效利用主存空间；缺点：空间分配麻烦；容易留下许多外碎片，造成存储空间利用率低下
2. 地址映射过程：

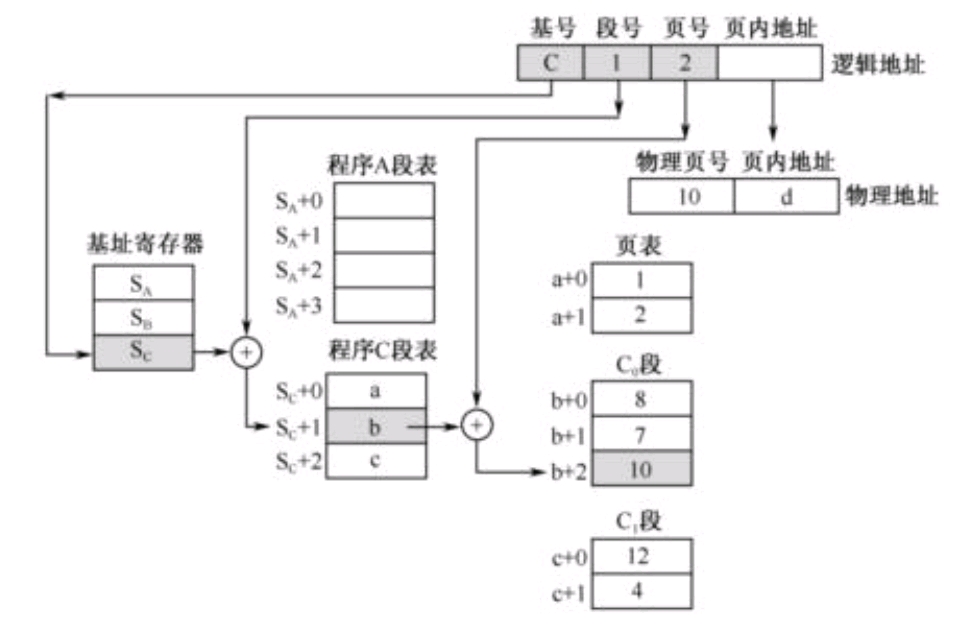






1. 段页式虚拟存储器
2. 实存被等分成页。每个程序先按照逻辑分段，每段再按照实存的页的大小分页（每一段保存是在一个页表中的，不同的段保存在不同的页表中），程序按页进行调入调出操作，但可按段进行编程、保护和共享。
3. 缺点：由于虚地址向主存地址的映射过程中需要多次查表，因而实现复杂度较高
4. 每个程序均通过一个段表+多个页表进行两级再定位
5. 虚地址的构成

|  |  |  |  |
| --- | --- | --- | --- |
| 基号 | 段号 | 段内的逻辑页号 | 页内的地址偏移量 |



总述：

大题：

1. 存储器的字位扩展，画图
2. 存储器的性能指标（容量、带宽）+多模块交叉存储器（顺序、交叉）求解存储器带宽
3. Cache的性能指标的计算（命中率，平均访问时间，访问效率）
4. Cache和主存之间的地址映射（全、直接、组）
5. 虚拟存储器的地址管理方式：页，段，段页