

《计算机组成原理》

（第四讲习题答案）

厦门大学信息学院软件工程系 曾文华

2022年3月29日

第4章 存储系统

- 4.1 存储器概述
- 4.2 半导体存储器
- 4.3 主存的组织及与CPU 的连接
- 4.4 并行主存系统
- 4.5 高速缓冲存储器
- 4.6 虚拟存储器

习题 (P147-152)

- 4.1
- 4.2
- 4.3
- 4.6
- 4.8
- 4.11
- 4.13
- 4.14
- 4.17

习题答案 (P147-152)

• 4.1 解释下列名称

- **存取时间**：也称存储器的访问时间，指启动一次存储器操作（读或写）到该操作完成所经历的时间；读时间和写时间可能不同，例如**DRAM**读慢写快，闪存（**U盘**）读快写慢。
- **存取周期**：指连续启动两次访问操作之间的最短时间间隔；存取周期 = 存取时间 + 恢复时间（存储器状态的稳定恢复时间）；存取周期 > 存取时间。
- **存储器带宽**：指单位时间内存储器所能传输的信息量；存储器带宽的单位：位/秒，**b/s**；字节/秒，**B/s**；存储器带宽与存取时间的长短和一次传输的数据位的多少有关；存取时间越短、数据位越宽，存储器带宽越高。
- **存储单元**：也称为存储元，存放**1**位二进制数。
- **边界对齐的数据存放**：假设计算机的字长为**32**位（即**1**个字=**32**位=**4**字节），则边界对齐要求：双字数据（**64**位，**8**字节）起始字节地址最末**3**位为**000**（**8**的倍数）；单字数据（**32**位，**4**字节）起始字节地址最末**2**位为**00**（**4**的倍数）；半字数据（**16**位，**2**字节）起始字节地址最末**1**位为**0**（**2**的倍数）；单字节数据（**8**位，**1**字节）起始字节地址可以任意。
- **大端存储**：**Big-Endian**，存储器的低字节地址单元中存放的是数据的最高字节。采用大端方式更符合人的思维习惯。
- **小端存储**：**Little-Endian**，存储器的低字节地址单元中存放的是数据的最低字节。采用小端方式则有利于计算机处理。

- **4.1 解释下列名称（续）**

- **静态存储器**：存储体以静态MOS存储元为基本单元组成的存储器称为静态MOS存储器，也称为静态随机存取存储器（SRAM: Static Random Access Memory）；静态MOS存储器（SRAM）存储1位二进制数需要6个MOS管，其存储密度较低。
- **动态存储器**：存储体以动态MOS存储元为基本单元组成的存储器称为动态MOS存储器，也称为动态随机存取存储器（DRAM: Dynamic Random Access Memory）；动态MOS存储器（DRAM）则采用数量较少的MOS管（如4个MOS管、3个MOS管、1个MOS管）和存储电容来存储1位二进制数，其存储密度较高。
- **刷新**：因为DRAM电容上的电荷会逐渐泄露，数据只能保存较短的时间，为避免数据丢失，必须定期采用类似读操作的方式对存储单元补充电荷，这个过程称为刷新。
- **刷新周期**：信息存储到数据丢失之前的时间称为最大刷新周期；存储器实际完成两次完整刷新之间的时间间隔称为刷新周期，常见的刷新周期有2ms、4ms、8ms等。
- **字扩展**：字扩展又称容量扩展或地址总线扩展；例如，可以利用8个256Kx8位的SRAM芯片，构成1个2Mx8位的存储器。
- **位扩展**：字扩展又称容量扩展或地址总线扩展；例如，可以利用8个256Kx8位的SRAM芯片，构成1个2Mx8位的存储器。
- **多体交叉存储器**：包括高位多体交叉存储器和低位多体交叉存储器。高位多体交叉存储器的结构与存储器字扩展完全相同，也称为顺序编址模式；高位地址经过译码器用于选择不同的存储体，低位地址同时送各个存储体。低位多体交叉存储器的结构与存储器字扩展也基本相同，只是低位地址经过译码器用于选择不同的存储体，而高位地址同时送各个存储体；低位多体交叉也称为交叉编址模式。

• 4.1 解释下列名称（续）

- **高速缓冲存储器**：**SRAM**的特点是速度快、容量小、价格高，**DRAM**的特点是速度慢、容量大、价格低，因此计算机的主存（内存，内存条）通常采用**DRAM**，而不是采用**SRAM**。为了提高**CPU**访问主存的速度，通常在**CPU**与主存之间增加一个隐藏的小容量快速**SRAM**，称为**cache**（高速缓冲存储器）。因为**CPU**执行的程序具有较强的局部性，可以将主存中经常访问或即将访问的数据的副本调度到**cache**中，使得大部分数据访问可以在**cache**中进行，从而提升系统的性能。
- **双端口存储器**：有两个相互独立的端口（左口、右口）。当左右两个端口的地址不同时，两个端口使用各自的地址线、数据线和控制线，此时可以同时存储体的两个不同的存储单元进行访问（读写）。当左右两个端口的地址相同时，将发生冲突（因为存储体不允许同时对同一个存储单元进行读写操作）；此时由判断逻辑决定是哪一端先访问存储体，如左边的端口先访问存储体，然后将右边端口的**/Busy_R**置为低电平（**/Busy_R=0**，表示右边的端口不能访问存储体）；等左边的端口访问结束后，再将右边端口的**/Busy_R**置为高电平（**/Busy_R=1**，表示右边的端口可以访问存储体）。
- **相联存储器**：**CAM**，**Content Addressable Memory**；普通的存储器是按地址进行访问，而相联存储器则是按内容进行访问；相联存储器用于存放**cache**的查找表，其内部存储的基本数据是键值对（**key**，**value**）；相联存储器的输入不是地址，而是检索关键字**key**，输出是**key**对应的**value**值。
- **时间局部性**：是指当程序访问一个存储位置时，该位置在未来可能会被多次访问；如程序的循环结构和调用过程就很好地体现了时间局部性。
- **地址映射**：如何将主存地址空间映射到**cache**的地址空间？即主存的某一块将载入到**cache**的哪一块中？地址映射有3种方法：全相联映射（**Full Associative Mapping**），直接相联映射（**Direct Mapping**），组相联映射（**Set Associative Mapping**）。
- **直接相联映射**：主存的某一块只能映射到**cache**的固定块中。
- **全相联映射**：主存的某一块可以映射到**cache**的任意块中。
- **组相联映射**：主存的某一块只能映射到**cache**的固定组中，在该组中可以映射到任意块中。

• 4.1 解释下列名称（续）

- **命中率**：如果CPU要访问的数据在cache中，则称为命中（Hit）。如果CPU要访问的数据不在cache中，则称为数据缺失（Miss，也称为不命中），此时要将缺失数据从主存调入cache中才能访问数据。假设 N_c 为命中cache的次数， N_m 为从主存中访问信息的次数（即不命中的次数），则命中率 h （Hit Ratio）： $h = N_c / (N_c + N_m)$ 。
- **虚拟存储器**：虚拟存储器处于“主存-辅存”存储层次，是为了解决存储系统“存储容量大”的问题；cache处于“CPU-主存”存储层次，是为了解决存储系统“访问速度快”的问题。有了虚拟存储器，程序员可以使用虚拟地址（虚拟地址空间比主存的物理地址空间要大很多）进行编程，这样程序员的编程不再受实际主存空间大小的限制。虚拟存储器也充分利用了程序的局部性，采用按需加载的方式加载程序代码和数据；根据程序局部性原理，通常程序只需要加载很小一部分空间即可运行，这种方式避免了将程序全部载入内存，大大提高了主存的利用率。
- **页框号**：虚拟地址（VA）= 虚拟页号（VPN）+ 虚拟页偏移（VPO）；物理地址（PA）= 物理页号（PPN）+ 物理页偏移（PPO）；虚拟页号也称为虚页号；物理页号也称为页框号、实页号。
- **页表（慢表）**：虚拟地址到物理地址的映射，本质上就是将虚拟页号（虚页号）转换为物理页号（实页号）；页式虚拟存储器中虚拟地址到物理地址之间的转换是基于页表进行的。页表是一张保存虚拟页号和物理页号对应关系的查找表，是一个由若干个表项组成的数组。页表采用虚拟页号作为索引进行访问，每一个表项主要包括：有效位和物理页号（PPN），另外还包括：修改位、使用位、权限位等信息。
- **页表项**：页表中的每一行称为页表项（PTE：Page Table Entry）。
- **TLB（快表）**：为了进一步提高虚拟存储器地址转换的速度，现代计算机中都设置一个转换旁路缓冲区（Translation Look-aside Buffer，TLB），用于缓冲经常访问的页表项（PTE，即页表中经常会被访问的行）。TLB本质上就是一个容量较小的cache，为提高查找速度，大多采用全相联或组相联方式，且采用随机替换算法。TLB也称为快表，存放在主存中的页表称为慢表；快表是按内容访问的，慢表是按地址访问的。在进行地址转换时，往往同时查快表和慢表，如果查快表命中，则从快表中得到物理页号，同时终止查慢表的过程；如果查快表不命中，则从慢表中得到物理页号。

• 4.1 解释下列名称（续）

- **LRU算法**：近期最少使用：Least Recently Used，LRU算法将近期内最久未被访问过的cache行替换掉。cache的每行设置一个计数器，当访问某cache行时，该cache行的计数器清0，其它cache行的计数器加1；因此计数器是用于统计未被访问的次数。当需要替换时，比较所有可替换行的计数值，将计数值最大的行替换掉（即近期最久未被访问的行）。
- **LFU算法**：最不经常使用：Least Frequently Used，LFU。LFU算法将访问次数最少的cache行（数据块）替换掉。cache的每行设置一个计数器，每访问该行一次，计数器加1；当需要替换时，对所有可替换行的计数值进行比较，将计数值最小的行替换掉（即最不经常使用的行）。
- **cache一致性**：因为cache是主存一部分内容的副本，在写入cache时，需要保证cache与主存数据的一致性；即cache的内容改变了，主存相应的内容也要改变。cache的写入策略主要有：写回法、写穿法。
- **写回法**：采用写回法，当写入cache时，只修改cache的内容，并不立即修改主存的内容；只有当该cache行被替换出去时，才将脏数据（cache被修改后的数据称为脏数据）写回主存。写回法可以使cache在大部分时间（没有发生替换的时间）的写入速度等于访问cache的速度；只有在发生替换时，cache的写入速度才等于访问主存的速度（此时要执行写入主存的操作）。采用写回法时，cache的每一行必须配置一个修改位，也称为脏位（Dirty Bit）；若该行被修改，则脏位=1，否则为0；当该行被替换出去时，如果脏位=1，则需要将该行的内容写入主存。
- **写穿法**：写穿法也称为直写法，当写入cache时，同时将该数据块也写回主存。写穿法的优点：cache的每行不需要设置一个修改位（脏位），并且发生替换时，直接将该cache行丢弃，不需要写回主存。写穿法的缺点：cache的写入时间将是访问主存的时间，从而大大降低写入cache的速度。写穿法较好地维护了cache与主存的一致性。

• 4.2 选择题

- (1) **A**
 - I: 正确
 - II: 正确
 - III: 错误
 - IV: 错误
 - 因此**仅I和II**
- (2) **A**
 - DRAM芯片容量=4Mx8位，地址线=22根，因为DRAM是采用地址复用技术，只需要一半的地址线=11根；数据线=8根，总数=11+8=19根
- (3) **D**
 - ROM容量=4KB，RAM容量=64KB-4KB=60KB；ROM芯片的数量=4KB/(2Kx8位)=2片，RAM芯片的数量=60KB/(4Kx4位)=30片

4.2 (3) 题目有误

“现要用2KBx8位的ROM芯片和4KBx4位的RAM芯片”，改为：“现要用2Kx8位的ROM芯片和4Kx4位的RAM芯片”

- (4) **D**
 - 2K的地址范围：0000H ~ 07FFH
 - 8K的地址范围：0000H ~ 07FFH，0800H ~ 0FFFH，1000H ~ 17FFH，1800H ~ 1FFFH
 - 0B1FH在0800H ~ 0FFFH范围内，其最小地址是0800H
- (5) **C**
 - A: 行地址=11根，列地址=0根，DRAM的地址线=max(行地址，列地址)=11根
 - B: 行地址=6根，列地址=5根，DRAM的地址线=max(行地址，列地址)=6根
 - C: 行地址=5根，列地址=6根，DRAM的地址线=max(行地址，列地址)=6根
 - D: 行地址=0根，列地址=11根，DRAM的地址线=max(行地址，列地址)=11根
 - B、C的地址线相同，但是C的行数少，因此C的刷新开销小 (r=32、c=64)
- (6) **B**
 - 存储器总线的带宽=总线宽度/时钟周期=总线宽度x工作频率=64位x1333MHz=10.664GB/s
 - 因为是3通道，因此存储器总线的带宽=3x10.664=31.992GB/s=32GB/s

• 4.2 选择题（续）

– (7) D

- 存储体0的地址为：8000、8004、8008；存储体1的地址为：8001、8005；存储体2的地址为：8002、8006；存储体3的地址为：8003、8007
- 主存地址序列为：8005、8006、8007、8008、8001、8002、8003、8004、8000
- A：8004和8008虽然都在存储体0，但是相差4个序列，因此不会冲突
- B：8002和8007分别在存储体2和存储体3中，因此不会冲突
- C：8001和8008分别在存储体1和存储体0中，因此不会冲突
- D：8000和8004都在存储体0，并且是先后访问，因此会发生冲突

– (8) B

- SRAM：不需要刷新；SDRAM：需要刷新；ROM：不需要刷新；FLASH：不需要刷新

– (9) B

- EPROM、DRAM、SRAM都属于随机存取方式的存储器；CDROM是光盘，不属于随机存取方式的存储器

– (10) A

- A：错误，Flash Memory的读、写速度不一样
- B：正确
- C：正确
- D：正确



	NAND Flash	NOR Flash
芯片容量	<32GBit	<1GBit
访问方式	顺序读写	随机读写
接口方式	任意I/O口	特定完整存储器接口
读写性能	读取快（顺序读） 写入快 擦除快（可按块擦除）	读取快（RAM方式） 写入慢 写入慢
使用寿命	百万次	十万次
价格	低廉	高昂

• 4.2 选择题（续）

– (11) A

- 数组a中通常包含若干个数据，这些数据在内存中是顺序存放的，因此具有**空间局部性**；数组a通常在程序的循环体中被执行若干次，因此还具有**时间局部性**

– (12) C

- cache共有16块，二路组相联，因此cache有 $16/2=8$ 组；主存块大小=32B，因此块内偏移=5位
- $129_{十进制}$ 号主存单元的地址= $1000\ 0001_{二进制}$ = $100\ 00001_{二进制}$ ，其对应的主存块号= $100_{二进制}=4$ ，将载入cache的**第4组**

– (13) A

- cache有4行，二路组相联，因此cache有 $4/2=2$ 组
- cache与主存之间交换的块大小为1个字，计算机按字编址，主存地址即为主存的块号
- **命中次数=1**



主存块号	0	4	8	2	0	6	8	6	4	8
对应cache组号	0	0	0	0	0	0	0	0	0	0
对应cache行号	0或1	0或1	0或1	0或1	0或1	0或1	0或1	0或1	0或1	0或1
cache 第0组、第0行	0^0	0^1	8^0	8^1	0^0	0^1	8^0	8^1	4^0	4^1
cache 第0组、第1行		4^0	4^1	2^0	2^1	6^0	6^1	6^0	6^1	8^0
cache 第1组、第0行										
cache 第1组、第1行										
命中情况	载入	载入	替换	替换	替换	替换	替换	命中	替换	替换

• 4.2 选择题（续）

— (14) C

- 主存地址=区地址+行索引+块内偏移=32位
- cache的数据容量=4K字，块大小=4字，因此cache有1K=1024行，行索引=10位
- 1个块=4个字=4x32位=16字节，块内偏移=4位
- 因此区地址=32-10-4=18位
- 因为采用写回的方式，因此cache每一行包括：主存数据块副本、区地址、有效位、脏位，即：4x32位+18位+1位+1位=148位
- 因此cache总容量=1024行x148位=148K位

— (15) D

- 指令cache和数据cache分离，CPU可以同时访问指令cache（取指令）和数据cache（取数据或写回数据），从而减少指令流水线的资源冲突

— (16) B

- 当采用页式虚拟存储器时，并配有TLB，最短的路径是TLB命中、cache命中，此时虽然不需要访问存储器，但是因为cache使用写穿的方式，“add xaddr, 3”指令需要完成写cache操作（x+3 -> x），同时要执行写存储器的操作（写穿方式），因此访问主存的次数至少是1次

— (17) D

- A: 第5种情况
- B: 第3种情况
- C: 第2种情况
- D: 第7种情况，不可能

序号	TLB	页	cache	可能性	说明
1	命中	命中	命中	可能	TLB命中则页一定命中，页载入主存和数据块载入cache并不同步，所以cache有可能命中，也有可能缺失
2	命中	命中	缺失		
3	缺失	命中	命中	可能	TLB缺失后还可以访问慢速页表（慢表），页载入主存和数据块载入cache并不同步，所以cache有可能命中，也有可能缺失
4	缺失	命中	缺失		
5	缺失	缺失	缺失	可能	这是最糟糕的情况，虚存系统初始化时常见
6	命中	缺失	缺失	不可能	页缺失说明页不在主存中，TLB中一定没有对应页表项，TLB不可能命中
7	命中	缺失	命中		
8	缺失	缺失	命中	不可能	页缺失说明数据也不在主存中，所以cache不可能命中

- 4.2 选择题（续）

- (18) A

- 主存地址空间大小=256MB，主存地址=28位
 - 虚拟地址空间大小=4GB，虚拟地址=32位
 - 页面大小=4KB，页偏移=12位
 - 物理地址 = 实页号（页框号）+ 页偏移，因此：页框号=28-12=16位
 - 虚拟地址 = 虚页号 + 页偏移，因此：虚页号=32-12=20位
 - 虚拟地址03FFF180H=虚页号 + 页偏移=03FFFH + 180H
 - 虚页号=03FFFH，查页表，得到页框号=0153H，有效位=1，表示命中，对应的主存地址= 0153H + 180H = 0153180H

- (19) D

- A: 正确
 - B: 正确
 - C: 正确
 - D: 错误：缺页处理程序返回到原来的进程，驱使引起缺页的指令重新启动

• 4.3 简答题

– (1) 计算机系统中采用层次化存储体系结构的目的是什么？

– 答：

- 存储系统层次结构利用程序局部性的原理，从系统级角度将速度、容量、成本各异的存储器有机组合在一起，全方位优化存储系统的各项性能指标；上层存储器可以为下层存储器做缓冲，将经常使用数据的副本调度到上层，这样CPU只需要访问上层快速的小容量存储器，即可获得大部分数据。

– (2) 为什么在存储器芯片中设置片选输入端？

– 答：

- 单片存储芯片的容量有限，通常需要将多片存储芯片按照一定的方式组织起来，并与CPU连接，从而获得一个大容量的存储器。通过设置片选端，可以由地址线经译码电路得到片选信号，从而选择不同的存储器芯片。

– (3) 动态MOS存储器为什么要刷新？如何刷新？

– 答：

- 因为DRAM电容上的电荷会逐渐泄露，数据只能保存较短的时间，为避免数据丢失，必须定期采用类似读操作的方式对存储单元补充电荷，这个过程称为刷新。
- 动态MOS存储器采用类似读操作的方式进行刷新。DRAM的刷新是按行进行的，每次完成1行的刷新，刷新操作只需要给出存储器的行地址，不需要给出存储器的列地址。常见的刷新方式：集中刷新方式；分散刷新方式；异步刷新方式。

– (4) 试述多体交叉存储器的设计思想和实现方法。

– 答：

- 高位多体交叉存储器的结构与存储器字扩展完全相同，也称为顺序编址模式；高位地址经过译码器用于选择不同的存储体，低位地址同时送各个存储体。
- 低位多体交叉存储器的结构与存储器字扩展也基本相同，只是低位地址经过译码器用于选择不同的存储体，而高位地址同时送各个存储体；低位多体交叉也称为交叉编址模式。低位多体交叉存储器的优点：由于程序具有局部性和连续性的特点，如果将程序的指令和数据分布在不同的存储体中，可以实现多个存储体并行工作。

– (5) 为什么说cache对程序员是透明的？

– 答：

- cache的数据查找、地址映射、替换策略、写入策略都是由cache控制器实现的，不需要操作系统干预。对程序员来说，在编写程序时，是按照主存的地址进行访存的，感觉不到计算机中有cache存在。因此cache对程序员来说是透明的。

• 4.3 简答题（续）

– （6）直接相联映射方式下，为什么不需要使用替换算法？

– 答：

- 直接相联映射方式，主存的某一块将映射到cache的固定块；当cache满时，只能替换掉该主存块对应的cache块，因此不需要替换算法。

– （7）为什么要考虑cache的一致性？

– 答：

- 因为cache是主存一部分内容的副本，在写入cache时，需要保证cache与主存数据的一致性；即cache的内容改变了，主存相应的内容也要改变。cache的写入策略主要有：写回法、写穿法。采用写回法，当写入cache时，只修改cache的内容，并不立即修改主存的内容；只有当该cache行被替换出去时，才将脏数据（cache被修改后的数据称为脏数据）写回主存。写穿法也称为直写法，当写入cache时，同时将该数据块也写回主存。写穿法的优点：cache的每行不需要设置一个修改位（脏位），并且发生替换时，直接将该cache行丢弃，不需要写回主存。操作的速度。

– （8）替换算法有哪几种？它们各有何优点？

– 答：

- 常用的替换算法有4种：先进先出算法（FIFO），最不经常使用算法（LFU），近期最少使用算法（LRU），随机替换算法。
- FIFO算法按照数据块进入cache的先后决定替换的顺序，在需要进行替换时，选择最先被载入cache的行（数据块）进行替换。FIFO算法的优点：系统开销较小。FIFO算法的缺点：没有考虑程序访问的局部性，可能会把一些需要经常使用的块（如循环程序块）也作为最早进入cache的块而替换掉，从而导致cache的命中率不高。
- LFU算法将访问次数最少的cache行（数据块）替换掉。LFU算法的不足：计数器记录的是cache工作后的历史访问统计情况，并不能严格反映近期访问情况。
- LRU算法的优点：保护了刚载入cache的新数据，符合cache的工作原理（程序局部性），使cache具有较高的命中率；LRU算法的缺点：需要快速比较多个cache行的计数器值，硬件实现比较复杂。
- 随机替换就是在需要进行替换时，从所有可替换的行中，随机选取一行进行替换。优点是实现最容易，而且替换速度也比前面三种算法快；缺点是随机替换的数据，很可能马上又要使用，从而降低cache的命中率和工作效率。

- **4.4** 对于**32KB**容量的存储器，若按**16位**字编址，其地址寄存器应是多少位？数据寄存器是多少位？

- 解：

- $32\text{KB}/16\text{位}=16\text{K}=2^{14}$

- 地址寄存器=**14位**

- 数据寄存器=**16位**

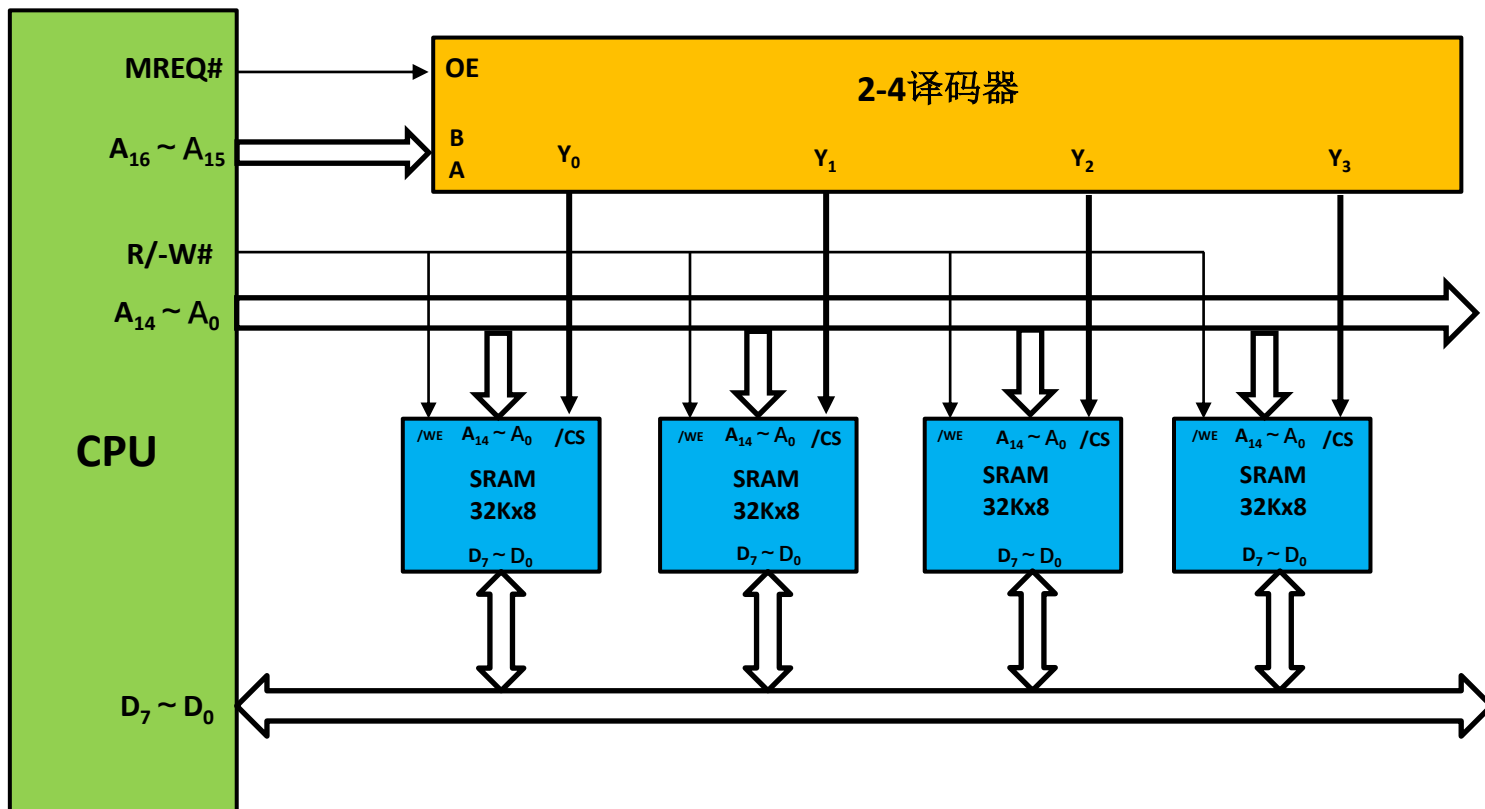
- **4.5 用4个32Kx8位SRAM存储芯片，可设计出哪几种不同容量和字长的存储器？画出相应设计图，并完成与CPU的连接。**

- **解：**

- **(1) 字扩展：32K -> 128K**
 - **4片32Kx8位 -> 128Kx8位**
- **(2) 位扩展：8位 -> 32位**
 - **4片32Kx8位 -> 32Kx32位**
- **(3) 字位扩展：32K -> 64K，8位 -> 16位**
 - **4片32Kx8位 -> 64Kx16位**

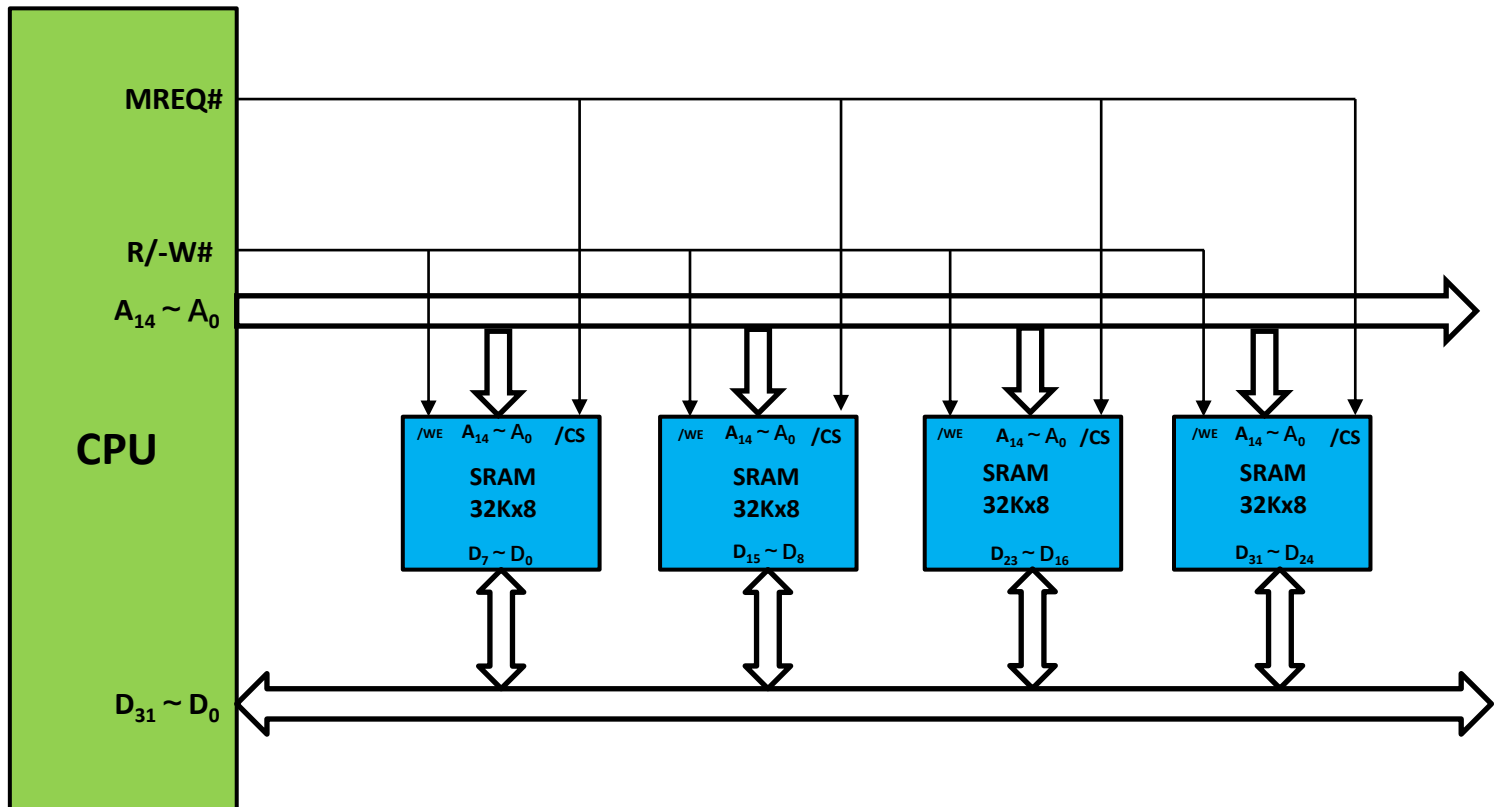
(1) 字扩展: 32K -> 128K

4片32Kx8位 -> 128Kx8位



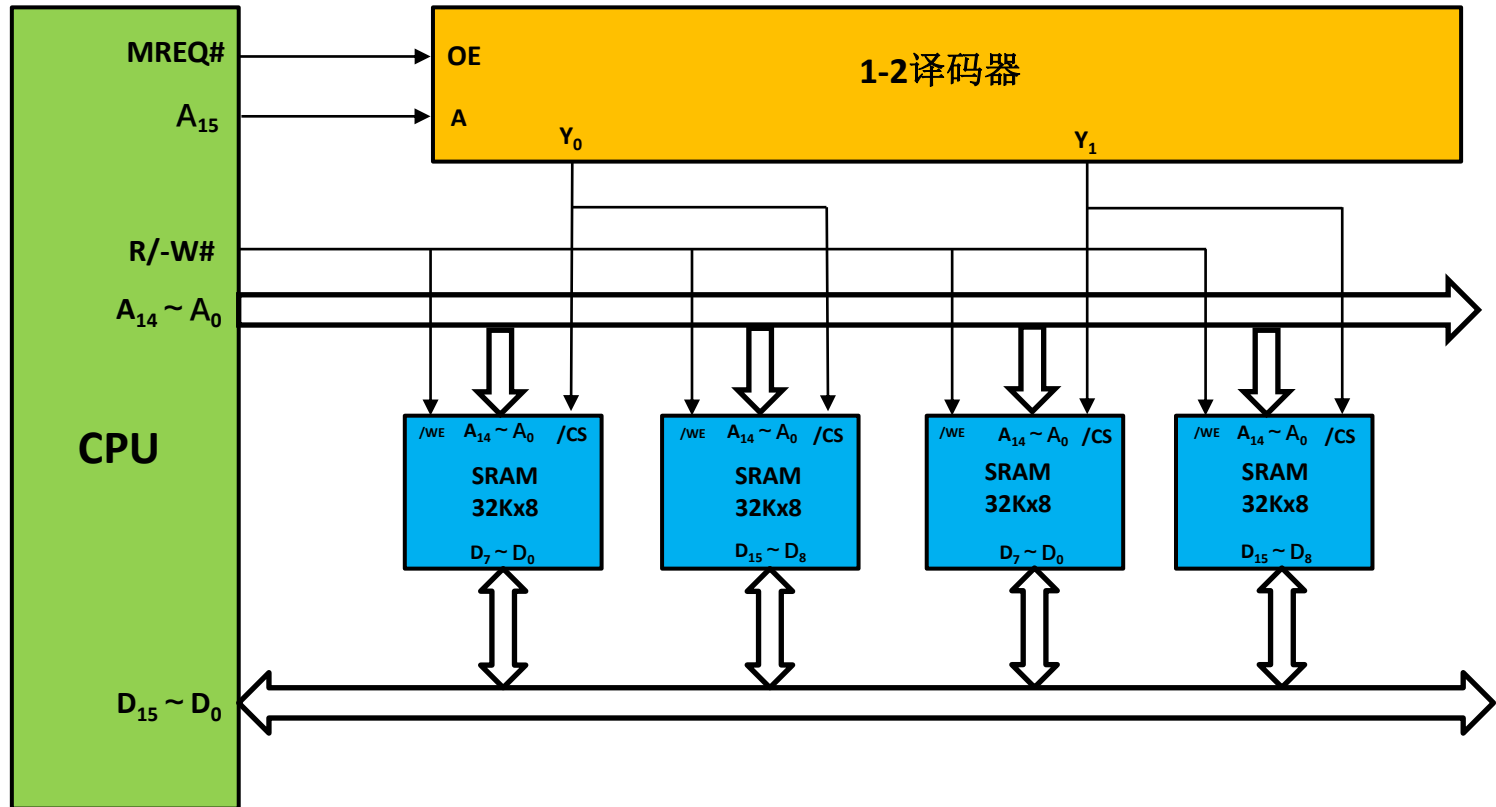
(2) 位扩展: 8位 -> 32位

4片32Kx8位 -> 32Kx32位



(3) 字位扩展: 32K -> 64K, 8位 -> 16位

4片32Kx8位 -> 64Kx16位



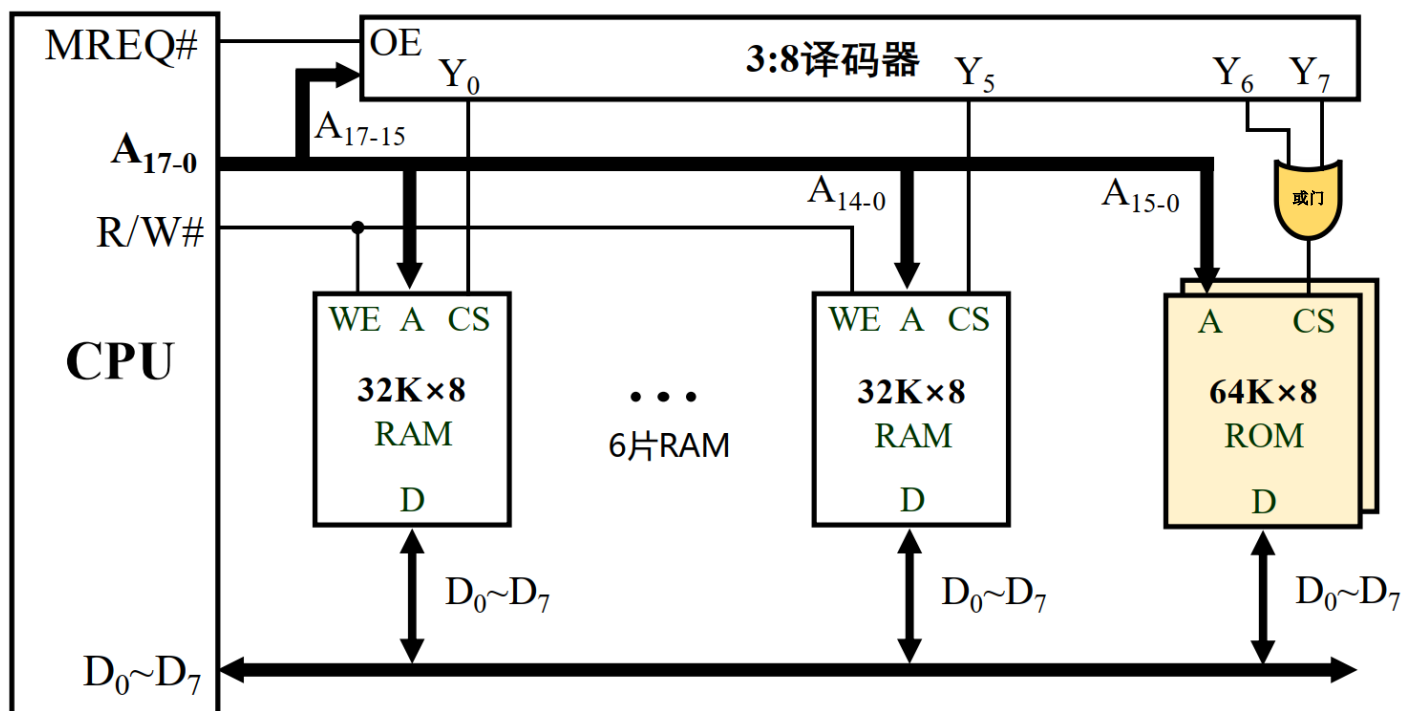
- **4.6 用32Kx8位RAM芯片和64Kx4位ROM芯片，设计256Kx8位存储器。其中，从30000H到3FFFFH的地址空间为只读存储器，其他为可读、可写存储器。完成存储器与CPU的连接。**

• **解：**

- 从30000H到3FFFFH的地址空间为只读存储器；因此，ROM的容量=64KB，需要2片64Kx4位ROM芯片
- RAM的容量=256KB-64KB=192KB，需要192/32=6片32Kx8位RAM芯片
- RAM采用位扩展，ROM采用字扩展
- 32Kx8位的RAM芯片，地址线=15根（ $A_{14} \sim A_0$ ），数据线=8根（ $D_7 \sim D_0$ ）
- 64Kx4位的ROM芯片，地址线=16根（ $A_{15} \sim A_0$ ），数据线=4根（ $D_3 \sim D_0$ ）
- 256KB的地址空间为：00000H ~ 3FFFFH，其中00000H ~ 2FFFFH为RAM空间，30000H ~ 3FFFFH为ROM空间

- 00 0000 0000 0000 0000 ~ 00 0111 1111 1111 1111 接第1片RAM
- 00 1000 0000 0000 0000 ~ 00 1111 1111 1111 1111 接第2片RAM
- 01 0000 0000 0000 0000 ~ 01 0111 1111 1111 1111 接第3片RAM
- 01 1000 0000 0000 0000 ~ 01 1111 1111 1111 1111 接第4片RAM
- 10 0000 0000 0000 0000 ~ 10 0111 1111 1111 1111 接第5片RAM
- 10 1000 0000 0000 0000 ~ 10 1111 1111 1111 1111 接第6片RAM
- 11 0000 0000 0000 0000 ~ 11 1111 1111 1111 1111 接2片ROM芯片

- 选用1个3-8译码器，将地址线的高3位（ $A_{17} \sim A_{15}$ ）接3-8译码器；译码器的输出为： $Y_7 \sim Y_0$
- 其中 $Y_5 \sim Y_0$ 分别接6个RAM芯片， $Y_7 \sim Y_6$ 分别通过一个或门接ROM芯片
- 地址线的低15位接6个RAM芯片，低16位接2个ROM芯片；8位数据线接6个RAM芯片，数据线的高4位、低4位分别接2个ROM芯片



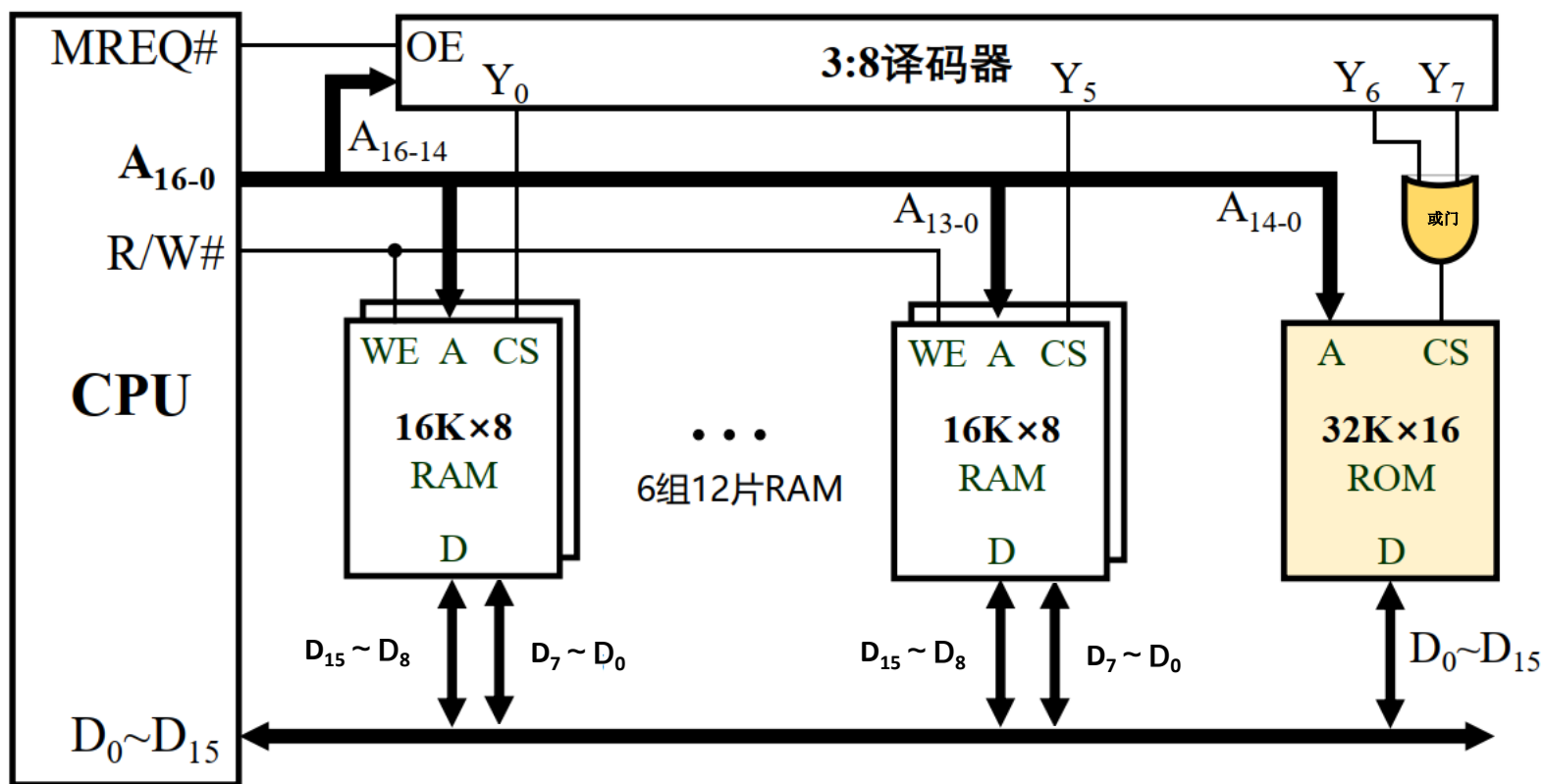
- 4.7 某计算机字长为16位，主存容量为128Kx16位，请用16Kx8位SRAM芯片和32Kx16位ROM芯片，为该计算机设计一个主存储器。要求18000H到1FFFFH为ROM区，其余为RAM区。画出存储器与CPU的连接。

4.7 题目有误
“主存容量为128KB”，改为：“主存容量为128Kx16位”

解：

- 18000H到1FFFFH为ROM区；因此，ROM的容量=32Kx16位，需要1片32Kx16位ROM芯片
- RAM区的容量=128Kx16位 - 32Kx16位=96Kx16位，需要(96Kx16)/(16Kx8)=12片16Kx8位RAM芯片
- RAM采用字位同时扩展，16K -> 96K，8位 -> 16位
- 32Kx16位的ROM芯片，地址线=15根 ($A_{14} \sim A_0$)，数据线=16根 ($D_{15} \sim D_0$)
- 16Kx8位的RAM芯片，地址线=14根 ($A_{13} \sim A_0$)，数据线=8根 ($D_7 \sim D_0$)
- 128Kx16位的地址空间为：00000H ~ 1FFFFH，其中00000H ~ 17FFFFH为RAM空间，18000H ~ 1FFFFH为ROM空间
- 0 0000 0000 0000 0000 ~ 0 0011 1111 1111 1111 接第1、2片RAM
- 0 0100 0000 0000 0000 ~ 0 0111 1111 1111 1111 接第3、4片RAM
- 0 1000 0000 0000 0000 ~ 0 1011 1111 1111 1111 接第5、6片RAM
- 0 1100 0000 0000 0000 ~ 0 1111 1111 1111 1111 接第7、8片RAM
- 1 0000 0000 0000 0000 ~ 1 0011 1111 1111 1111 接第9、10片RAM
- 1 0100 0000 0000 0000 ~ 1 0111 1111 1111 1111 接第11、12片RAM
- 1 1000 0000 0000 0000 ~ 1 1111 1111 1111 1111 接1片ROM芯片

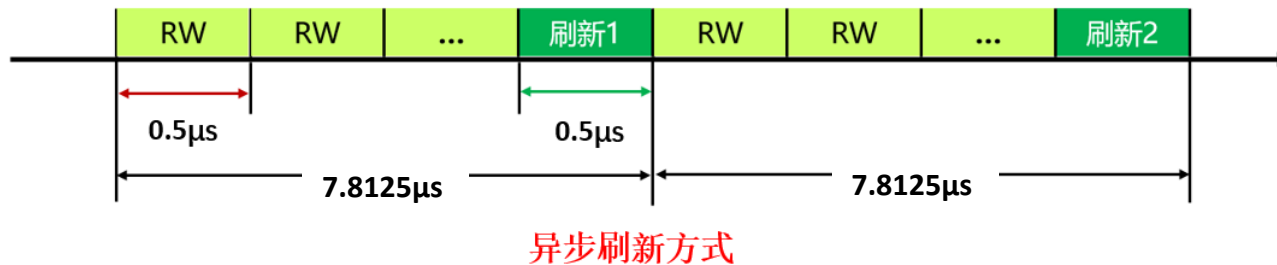
- 选用1个3-8译码器，将地址线的高3位（ $A_{16} \sim A_{14}$ ）接3-8译码器；译码器的输出为： $Y_7 \sim Y_0$
- 其中 $Y_5 \sim Y_0$ 分别接6个RAM芯片， $Y_7 \sim Y_6$ 分别通过一个或门接ROM芯片
- 地址线的低14位接12个RAM芯片，低15位接1个ROM芯片；16位数据线的高8位接6个RAM芯片、低8位接另外6个RAM，16位数据线直接接1个ROM芯片



- **4.8** 用64Kx1位的DRAM芯片构成1Mx8位的存储器，若采用异步刷新，每行刷新闻隔不超过2ms，则产生刷新信号的间隔时间是多少？假设读写周期为0.5μs，若采用集中刷新方式，则存储器刷新一遍最少需要多少个读写周期？CPU的“死”时间是多少？

解：

- DRAM芯片为64Kx1位，64K=65536=256X256，即存储体为256行、256列
- 采用异步刷新方式，刷新间隔=2ms/256行=7.8125 μs，即每隔7.8125 μs刷新一次，每次的刷新时间=读写周期= 0.5 μs



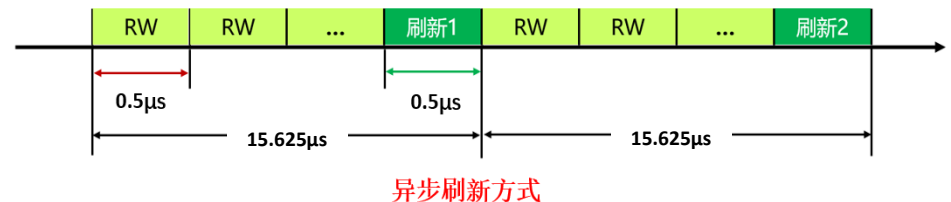
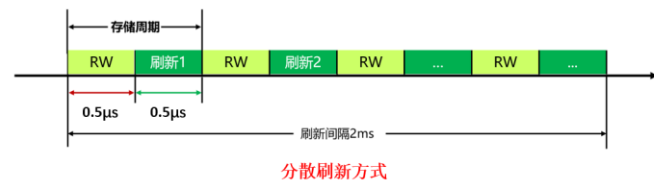
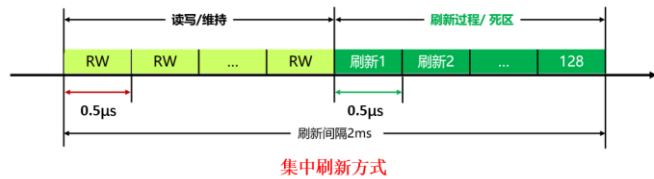
- 采用集中刷新方式，存储器刷新一遍需要256个读写周期（存储体有256行）
- 此时，CPU的“死”时间=256 x 0.5 μs =128 μs；死区率= 128 μs /2ms = 6.4 %

- 4.9 设有某动态RAM芯片，容量为64Kx1位，除电源线、接地线和刷新线外，该芯片的最小引脚数量是多少？
- 解：
 - 64Kx1位，地址线=16根（ $2^{16}=64K$ ），因为是DRAM，采用地址复用技术（行地址、列地址复用），实际只需要8根地址线（一半的地址线）
 - 数据线=1根
 - 此外，DRAM芯片还需要/RAS和/CAS控制信号，共2根
 - 因此该芯片的最小引脚数量是 $8+1+2=11$ 根

- 4.10 用16Kx1位的DRAM芯片构成64Kx8位的存储器，设存储器的读写周期为 $0.5\mu\text{s}$ ，要使CPU在1s内至少访问存储器一次，采用哪种刷新方式比较合适？若每行刷新间隔不超过2ms，该方式下刷新信号的产生周期是多少？

解：

- 因为集中刷新方式有“死区”，因此不能采用集中刷新方式；而采用异步刷新方式或分散刷新方式比较合适。
- DRAM芯片为16Kx1位， $16\text{K}=16384=128\times 128$ ，即存储体为128行、128列；采用异步刷新方式，刷新间隔= $2\text{ms}/128\text{行}=15.625\mu\text{s}$ ，即每隔 $15.625\mu\text{s}$ 刷新一次，每次的刷新时间=读写周期= $0.5\mu\text{s}$ ；因此异步刷新方式下刷新信号的产生周期是 $15.625\mu\text{s}$



- 4.11 设cache的容量为 2^{14} 块，每块是一个32位字，主存容量是cache容量的256倍，其中有表4.13所示的数据（地址和数据均采用十六进制表示）。分别采用全相联映射、直接相联映射、四路组相联映射方式，将主存中这些数据载入cache后，cache各块中的数据内容及相应的标志是什么？

表4.13 主存数据分布情况

地址	数据	地址	数据
000000	87568536	01FFFC	4FFFFC68
000008	87792301	FFFFF8	01BF2460
010004	9ABEFC00		

解：

- cache的容量为 2^{14} 块，cache块地址=14位
- 每块是一个32位字=4B，块内偏移=2位（ $2^2=4$ ）
- cache地址=14+2=16位
- 主存容量是cache的256倍，主存块为 256×2^{14} 块= 2^{22} 块，主存块地址=22位
- 主存地址=主存块地址+块内偏移=22+2=24位
- （1）全相联映射：**
 - 主存地址=主存块地址（tag）+块内偏移（offset）=22位 + 2位
 - cache每一行的内容=有效位+主存块地址（tag）+数据块副本
 - 例如表4.13中的第一个数据块：主存地址=000000H，主存块地址=主存地址/4=000000H，数据块副本=87568536H，有效位=1，假设映射到cache的第0行；其他行类似得到

全相联映射方式cache每行的内容

cache行	有效位	主存块地址（tag）（22位）	数据块副本（32位）
0	1	000000	87568536
1	1	000002	87792301
2	1	004001	9ABEFC00
3	1	007FFF	4FFFFC68
4	1	3FFFFE	01BF2460

- 4.11 设cache的容量为 2^{14} 块，每块是一个32位字，主存容量是cache容量的256倍，其中有表4.13所示的数据（地址和数据均采用十六进制表示）。分别采用全相联映射、直接相联映射、四路组相联映射方式，将主存中这些数据载入cache后，cache各块中的数据内容及相应的标志是什么？

表4.13 主存数据分布情况

地址	数据	地址	数据
000000	87568536	01FFFC	4FFFFC68
000008	87792301	FFFFF8	01BF2460
010004	9ABEFC00		

- 解（续）：
 - cache的容量为 2^{14} 块，cache块地址=14位
 - 每块是一个32位字=4B，块内偏移=2位（ $2^2=4$ ）
 - cache地址=14+2=16位
 - 主存容量是cache的256倍，主存块为 256×2^{14} 块= 2^{22} 块，主存块地址=22位
 - 主存地址=主存块地址+块内偏移=22+2=24位
 - （2）直接相联映射：
 - 主存地址=区地址（tag）+行索引（index）+块内偏移（offset）= 8位 + 14位 + 2位
 - cache每一行的内容=有效位+区地址（tag）+数据块副本
 - 例如表4.13中的第一个数据块：主存地址=000000H，区地址=00H，行索引=0000H，数据块副本=87568536H，有效位=1，映射到cache的第0000H行；其他行类似得到

直接相联映射方式cache每行的内容

cache行	有效位	区地址（tag）（8位）	数据块副本（32位）
0000	1	00	87568536
0002	1	00	87792301
0001	1	01	9ABEFC00
3FFF	1	01	4FFFFC68
3FFE	1	FF	01BF2460

- 4.11 设cache的容量为 2^{14} 块，每块是一个32位字，主存容量是cache容量的256倍，其中有表4.13所示的数据（地址和数据均采用十六进制表示）。分别采用全相联映射、直接相联映射、四路组相联映射方式，将主存中这些数据载入cache后，cache各块中的数据内容及相应的标志是什么？

表4.13 主存数据分布情况

地址	数据	地址	数据
000000	87568536	01FFFC	4FFFFC68
000008	87792301	FFFFF8	01BF2460
010004	9ABEFC00		

- 解（续）：
 - cache的容量为 2^{14} 块，cache块地址=14位
 - 每块是一个32位字=4B，块内偏移=2位（ $2^2=4$ ）
 - cache地址=14+2=16位
 - 主存容量是cache的256倍，主存块为 256×2^{14} 块= 2^{22} 块，主存块地址=22位
 - 主存地址=主存块地址+块内偏移=22+2=24位
 - （3）四路组相联映射：
 - 主存地址=标记（tag）+组索引（index）+块内偏移（offset）= 10位 + 12位 + 2位
 - cache每一行的内容=有效位+标记（tag）+数据块副本
 - 例如表4.13中的第一个数据块：主存地址=000000H，标记=000H，组索引=000H，数据块副本=87568536H，有效位=1，映射到cache的第000H组；其他行类似得到

四路组相联映射方式cache每行的内容

cache组	有效位	标记（tag）（10位）	数据块副本（32位）
000	1	000	87568536
002	1	000	87792301
001	1	004	9ABEFC00
FFF	1	007	4FFFFC68
FFE	1	3FF	01BF2460

- **4.12** 某计算机的cache由64个存储块构成，采用四路组相联映射方式，主存包含4096个存储块，每块由128个字组成，访问地址为字地址。（1）主存地址和cache地址各有多少位？（2）按照题干条件中的映射方式，列出主存地址的划分情况，并标出各部分的位数。

• 解：

— （1）

- cache由64个存储块构成，cache的块地址=6位（ $2^6=64$ ）；采用四路组相联映射方式，cache有16组，组地址=4位
- 每块由128个字组，块内偏移=7位（ $2^7=128$ ）
- cache地址=6+7=13位

- 主存包含4096个存储块，主存块地址=12位（ $2^{12}=4096$ ）
- 主存地址=主存块地址+块内偏移=12+7=19位

— （2）

- 四路组相联映射：主存地址=标记（tag）+组索引（index）+块内偏移（offset）= 8位 + 4位 + 7位

- **4.13** 某计算机的主存容量为4MB，cache容量为16KB，每块包含8个字，每字为32位，映射方式采用四路组相联。设cache的初始状态为空，CPU依次从主存第0、1、2、...、99号单元读出100个字（每次读一个字），并重复此操作10次，替换算法采用LRU算法。（1）求cache的命中率。（2）若cache比主存快10倍，分析采用cache后存储访问速度提高了多少？

解：

— (1)

- 主存容量为4MB，因此，主存地址=22位；cache容量为16KB，因此，cache地址=14位
- 每块包含8个字，每字为32位，因此，每块=8x32=32B，块内偏移=5位
- 主存块地址=22-5=17位，共217=131072块
- cache块地址=14-5=9位，共有512块；映射方式采用四路组相联，因此，cache有512/4=128组
- 因为每块包含8个字，因此，主存第0、1、2、...、99号单元，分布在主存的第0、1、2、...、12块中（13个块）；13个主存块没有超出cache的组，因此不需要执行替换
- 第1次读100个字时，只有读第0、8、16、24、32、40、48、56、64、72、80、88、96号单元时不命中（13次），其余87次都命中；后面重复操作时，全部命中；因此，1000次访问中，只有13次不命中，其余987次都命中，命中率=987/1000=98.7%

— (2)

- 采用cache后的平均访问时间： $t_a = ht_c + (1-h)t_m$ ； $t_m = 10t_c$
- 因此， $t_a = ht_c + (1-h)10t_c = 0.987t_c + 0.013 \times 10t_c = 1.117t_c$
- 不采用cache的访问时间=10t_c
- 用cache后存储访问速度提高了： $10t_c / 1.117t_c \approx 8.95$ 倍

主存数据块

	第0字	第1字	第2字	第3字	第4字	第5字	第6字	第7字
第0块	0	1	2	3	4	5	6	7
第1块	8	9	10	11	12	13	14	15
第2块	16	17	18	19	20	21	22	23
第3块	24	25	26	27	28	29	30	31
第4块	32	33	34	35	36	37	38	39
第5块	40	41	42	43	44	45	46	47
第6块	48	49	50	51	52	53	54	55
第7块	56	57	58	59	60	61	62	63
第8块	64	65	66	67	68	69	70	71
第9块	72	73	74	75	76	77	78	79
第10块	80	81	82	83	84	85	86	87
第11块	88	89	90	91	92	93	94	95
第12块	96	97	98	99				
...								
...								
...								
第131071块								

- **4.14** 假定某数组元素按行优先顺序存放在主存中，则在以下两段伪代码A和B中，分析下列问题：
（1）两段代码中对数组访问的时间局部性和空间局部性。（2）变量sum的时间局部性和空间局部性。（3）for循环体对指令访问的时间局部性和空间局部性。
- 解：
 - （1）
 - 程序A的数组a，其存储次序是按行优先顺序，访问顺序也是按行优先顺序，因此具有空间局部性；但是数组a的每个元素只使用一次，因此不具有时间局部性。
 - 程序B的数组a，其存储次序是按行优先顺序，但是访问顺序是按列优先顺序，因此不具有空间局部性；并且数组a的每个元素只使用一次，因此也不具有时间局部性。
 - （2）sum是单个变量，不存在空间局部性；但是变量sum在每次循环执行时都会用到，因此具有时间局部性。
 - （3）for循环体中的指令会被反复执行，因此具有时间局部性；此外for循环体中的指令序列通常会顺序执行，因此也具有一定的空间局部性。

程序A:

```
int sum_array_A(int a[M][N])
{
    int i,j,sum=0;
    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum;
}
```

程序B:

```
int sum_array_B(int a[M][N])
{
    int i,j,sum=0;
    for (i = 0; i < N; i++)
        for (j = 0; j < M; j++)
            sum += a[j][i];
    return sum;
}
```

- **4.15** 主存容量为**8MB**，虚存容量为**2GB**，分页管理时若页面大小为**4KB**，求出对应的**VPN**、**VPO**、**PPN**、**PPO**的位数。

- 解：

- 虚拟地址 (**VA**) = 虚拟页号 (**VPN**) + 虚拟页偏移 (**VPO**)
- 物理地址 (**PA**) = 物理页号 (**PPN**) + 物理页偏移 (**PPO**)

- 主存容量为**8MB**，因此，**PA=23**位

- 虚存容量为**2GB**，因此，**VA=31**位

- 分页管理时若页面大小为**4KB**，因此，**VPO=PPO=12**位

- **VPN=VA-VPO=31-12=19**位

- **PPN=PA-PPO=23-12=11**位

- 4.16 某页式虚拟存储器共8页，每页为1KB，主存容量为4KB，页表如表4.14所示。

表4.14 虚拟存储器页表

虚页号	0	1	2	3	4	5	6	7
实页号	3	2	1	2	3	1	0	0
载入位	1	1	0	0	1	0	1	0

- (1) 失效的页有哪几页？
- (2) 虚地址0、3028、1023、2048、4096、8000（均为十进制）的实地址分别是多少？

解：

- (1)
 - 表4.14中载入位=0的为失效页，即虚页号为2、3、5、7的页
 - (2)
 - 虚拟地址（VA）= 虚拟页号（VPN）+ 虚拟页偏移（VPO）；物理地址（PA）= 物理页号（PPN）+ 物理页偏移（PPO）
 - 每页为1KB，因此，VPO=PPO=10位；主存容量为4KB，因此，PA=12位
 - 页式虚拟存储器共8页，因此，VPN=3位；VA=VPN+VPO=3+10=13位；PPN=PA-PPO=12-10=2位
- ① 虚地址=0=000 00 0000 0000，因此，VPN=000=0，查表4.14，得到实页号=PPN=3=11，载入位=1，命中，实地址=11 00 0000 0000=C00H=3072
 - ② 虚地址=3028=010 11 1101 0100，因此，VPN=010=2，查表4.14，得到实页号=PPN=1=01，载入位=0，缺失
 - ③ 虚地址=1023=000 11 1111 1111，因此，VPN=000=0，查表4.14，得到实页号=PPN=3=11，载入位=1，命中，实地址=11 11 1111 1111=FFFH=4095
 - ④ 虚地址=2048=010 00 0000 0000，因此，VPN=010=2，查表4.14，得到实页号=PPN=1=01，载入位=0，缺失
 - ⑤ 虚地址=4096=100 00 0000 0000，因此，VPN=100=4，查表4.14，得到实页号=PPN=3=11，载入位=1，命中，实地址=11 00 0000 0000=C00H=3072
 - ⑥ 虚地址=8000=111 11 0100 0000，因此，VPN=111=7，查表4.14，得到实页号=PPN=0=00，载入位=0，缺失

- 4.17 某计算机系统有一个TLB和L1级数据cache，存储系统按字节编址，虚拟存储器容量为2GB，主存容量为4MB，页大小为128KB，TLB采用四路组相联方式，共有16个页表项。cache容量为16KB，每块包含8个字，每字为32位，映射方式采用四路组相联，回答下列问题：
- (1) 虚拟地址中哪几位表示虚拟页号？哪几位表示页内地址？虚拟页号中哪几位表示TLB标记？哪几位表示TLB索引？
- (2) 物理地址中哪几位表示物理页号？哪几位表示偏移地址？
- (3) 为实现主存与数据cache之间的组相联映射，对该地址应进行怎样的划分？

• 解：

– (1)

- 虚拟地址 (VA) = 虚拟页号 (VPN) + 虚拟页偏移 (VPO)
- 物理地址 (PA) = 物理页号 (PPN) + 物理页偏移 (PPO)

- 虚拟存储器容量为2GB，因此，VA=31位
- 主存容量为4MB，因此，PA=22位
- 页大小为128KB，因此，VPO=PPO=17位
- $VPN = VA - VPO = 31 - 17 = 14$ 位
- $PPN = PA - PPO = 22 - 17 = 5$ 位

• 虚拟页号=14位；页内地址=17位

- TLB采用四路组相联方式， $VPN = \text{TLB标记} + \text{TLB索引}$
- TLB共有16个页表项，四路组相联，有 $16/4=4$ 组，因此，TLB索引=2位，TLB标记=14-2=12位

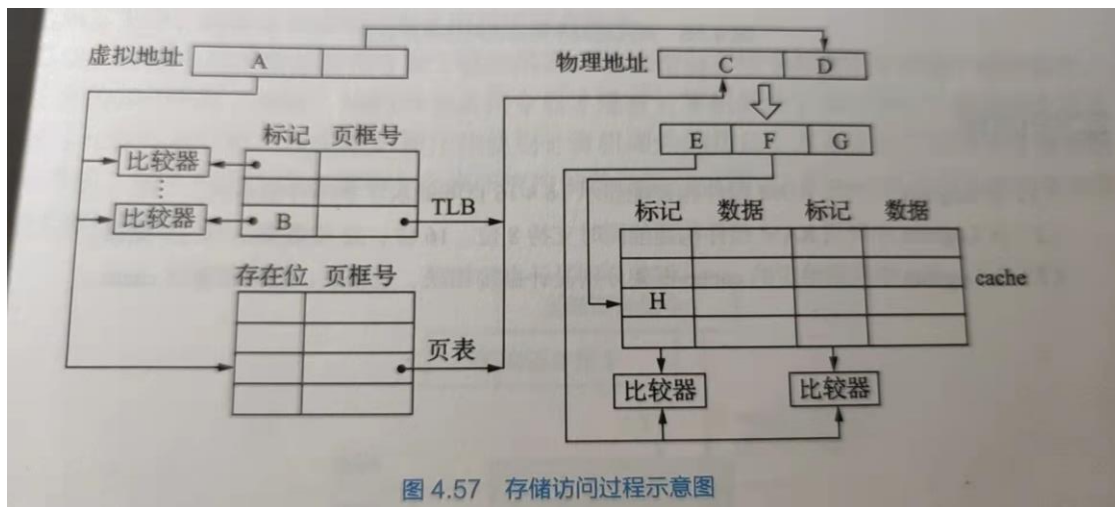
– (2)

• 物理页号=5位；偏移地址=17位

– (3)

- cache容量为16KB，因此，cache的地址=14位
- 每块包含8个字，每字为32位，因此，块地址=3+2=5位
- cache块地址=14-5=9位，共512块
- cache采用四路组相联，因此cache有 $512/4=128$ 组，组地址=7位
- 主存地址=22位=标记+组地址+偏移地址=10位+7位+5位

- 4.18 某计算机采用页式虚拟存储管理方式，按字节编址，虚拟地址为32位，物理地址为24位，页大小为8KB；TLB采用全相联映射；cache数据区大小为64KB，按二路组相联方式组织，主存块大小为64B。存储访问过程的示意图如图4.57所示。请回答下列问题：
- (1) 图中字段A~G的位数各是多少？TLB标记字段B中存放的是什么信息？
- (2) 将块号为4099的主存块装入cache中时，映射的cache组号是多少？对应H字段的内容是什么？
- (3) cache缺失处理的时间开销大，还是缺页处理的时间开销大？为什么？
- (4) 为什么cache可以采用写穿策略，而修改页面内容时总是采用写回策略？

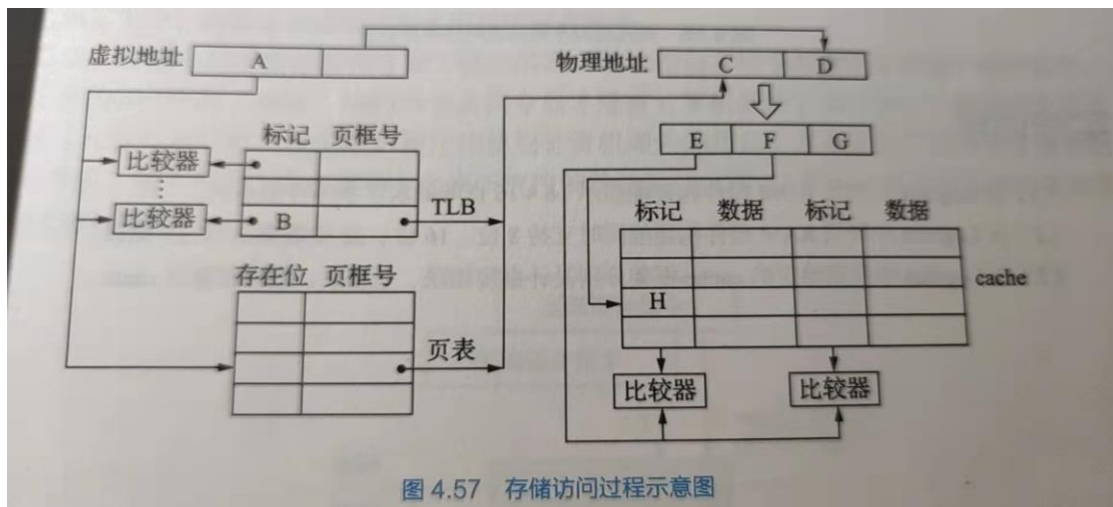


解：

— (1)

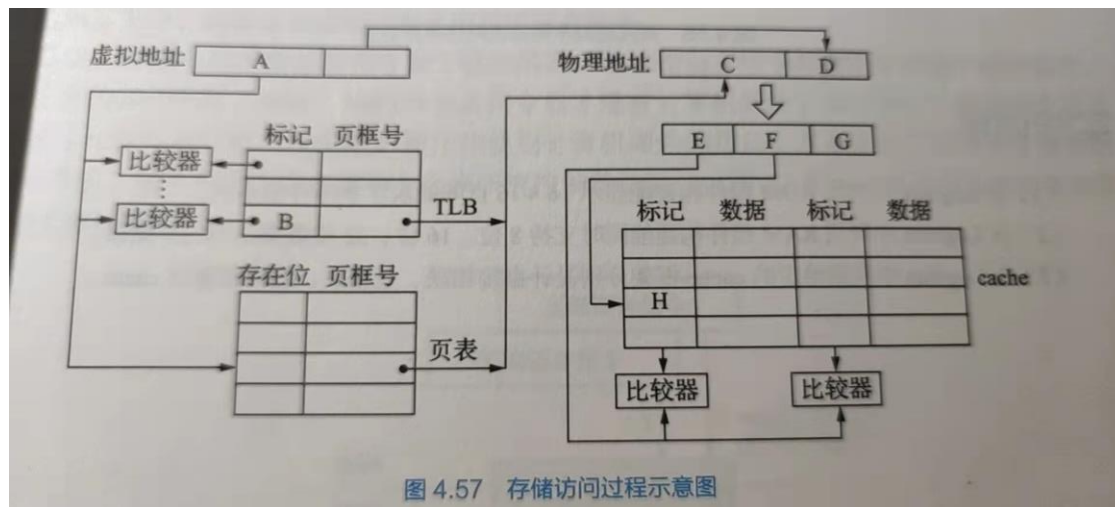
- 虚拟地址 (VA) = 虚拟页号 (VPN) + 虚拟页偏移 (VPO)
- 物理地址 (PA) = 物理页号 (PPN) + 物理页偏移 (PPO)
- 页大小为8KB，因此，VPO=PPO=13位；VPN=VA-VPO=32-13=19位；PPN=PA-PPO=24-13=11位
- A=VPN=19位，C=PPN=11位，D=PPO=13位
- TLB采用全相联映射，因此，TLB中的标记=B=VPN=19位
- 主存块大小为64B，因此偏移地址=6位=G
- cache数据区大小为64KB，因此，cache有64KB/64B=1K=1024行，二路组相联方式组织，cache有1024/2=512组
- 因此，F=cache的组数=9位
- E=24-9-6=9位
- A=19位，B=19位，C=11位，D=13位，E=9位，F=9位，G=6位
- TLB中标记字段B的内容是虚拟页号 (VPN)，表示该TLB项对应哪个虚页的页表项。

- 4.18 某计算机采用页式虚拟存储管理方式，按字节编址，虚拟地址为32位，物理地址为24位，页大小为8KB；TLB采用全相联映射；cache数据区大小为64KB，按二路组相联方式组织，主存块大小为64B。存储访问过程的示意图如图4.57所示。请回答下列问题：
- (1) 图中字段A~G的位数各是多少？TLB标记字段B中存放的是什么信息？
- (2) 将块号为4099的主存块装入cache中时，映射的cache组号是多少？对应H字段的内容是什么？
- (3) cache缺失处理的时间开销大，还是缺页处理的时间开销大？为什么？
- (4) 为什么cache可以采用写穿策略，而修改页面内容时总是采用写回策略？



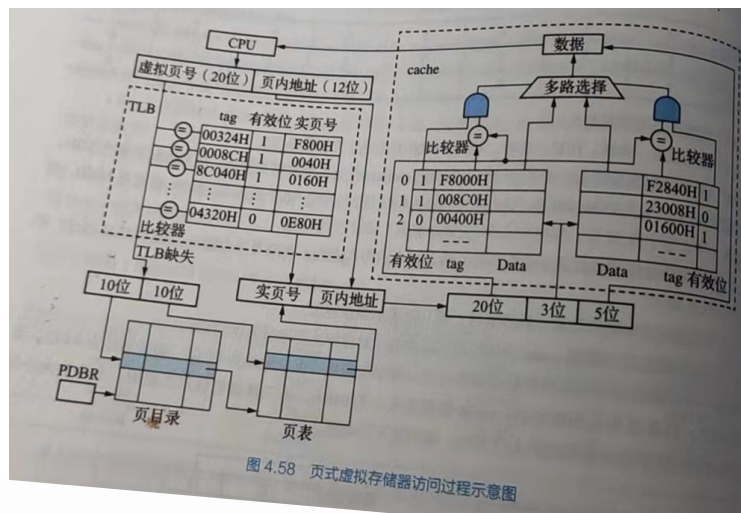
- 解（续）：
- (2)
- cache采用二路组相联映射方式，主存地址=标记（9位）+组索引（9位）+块内偏移（6位）=主存块地址（18位）+块内偏移（6位）
- 块号为4099（十进制）的主存块，块地址=4099=00001000 00000011
- 组索引=00000011=3
- 标记=00001000
- 映射的cache组号是3；对应H字段的内容是0 0000 1000

- 4.18 某计算机采用页式虚拟存储管理方式，按字节编址，虚拟地址为32位，物理地址为24位，页大小为8KB；TLB采用全相联映射；cache数据区大小为64KB，按二路组相联方式组织，主存块大小为64B。存储访问过程的示意图如图4.57所示。请回答下列问题：
- (1) 图中字段A~G的位数各是多少？TLB标记字段B中存放的是什么信息？
- (2) 将块号为4099的主存块装入cache中时，映射的cache组号是多少？对应H字段的内容是什么？
- (3) cache缺失处理的时间开销大，还是缺页处理的时间开销大？为什么？
- (4) 为什么cache可以采用写穿策略，而修改页面内容时总是采用写回策略？



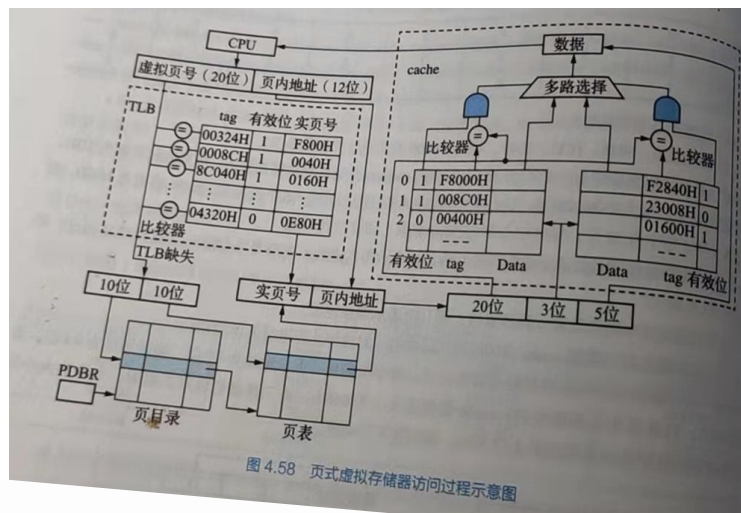
- 解（续）：
- (3)
 - 因为缺页处理需要访问磁盘，而cache缺失只要访问主存，cache缺失带来的开销小，而处理缺页的开销大。
- (4)
 - cache可以采用写穿策略时，需要同时写cache和写主存。
 - 修改页面内容时（即虚拟存储器），如果采用写穿策略，则需要同时写主存和写磁盘。
 - 而写磁盘比写主存慢得多；所以cache可以采用写穿策略，而虚存则应采用写回策略（不需要同时写磁盘）。

- 4.19 某计算机采用页式虚拟存储管理方式，按字节编址。CPU进行存储访问的过程如图4.57所示。请回答下列问题：
- (1) 主存的物理地址占多少位？
- (2) TLB采用什么映射方式？TLB是用SRAM还是DRAM实现？
- (3) cache采用什么映射方式？若cache采用LRU替换算法和写回策略，则cache每行中除数据（Data）、tag和有效位外，还应有哪些附加位？cache总容量是多少？cache中有效位的作用是什么？
- (4) 若CPU给出的虚拟地址为0008 C040H，则对应的物理地址是多少？是否在cache中命中？说明理由。若CPU给出的虚拟地址是0007 C260H，则该地址所在主存块映射到的cache组号是多少？



- 解：
- (1)
- 主存物理地址=20+3+5=28位
- (2)
- TLB采用全相联映射方式
- TLB用SRAM实现

- 4.19 某计算机采用页式虚拟存储管理方式，按字节编址。CPU进行存储访问的过程如图4.57所示。请回答下列问题：
- (1) 主存的物理地址占多少位？
- (2) TLB采用什么映射方式？TLB是用SRAM还是DRAM实现？
- (3) cache采用什么映射方式？若cache采用LRU替换算法和写回策略，则cache每行中除数据（Data）、tag和有效位外，还应有哪些附加位？cache总容量是多少？cache中有效位的作用是什么？
- (4) 若CPU给出的虚拟地址为0008 C040H，则对应的物理地址是多少？是否在cache中命中？说明理由。若CPU给出的虚拟地址是0007 C260H，则该地址所在主存块映射到的cache组号是多少？

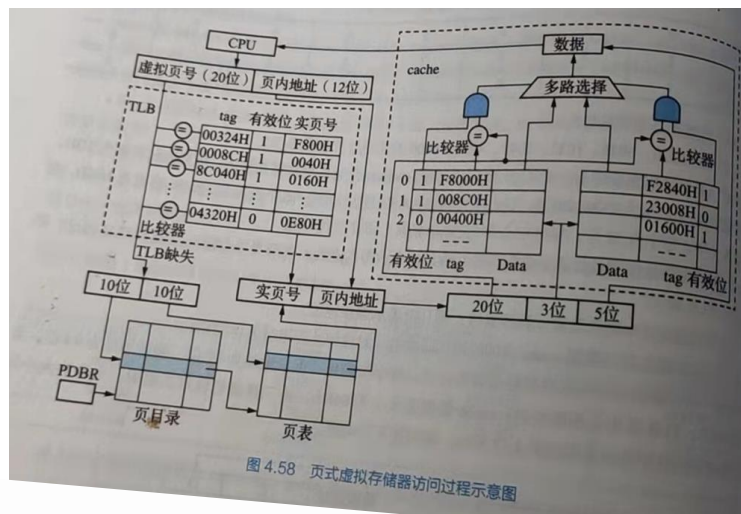


解（续）：

— (3)

- cache采用二路组相联映射方式
- cache采用LRU替换算法，因为是二路组相联映射方式，只需要在cache的每行中增加1位LRU位用于替换计数
- cache采用写回策略，则还应有脏位（Dirty Bit）
- 因此还应有2位附加位（LRU位、脏位）
- cache有8组，每组2行，共16行；每行包括：数据块、tag、有效位、LRU位、脏位
- 主存物理地址=20+3+5=28位，因此cache的块内地址=5位，即数据块=32x8
- tag=20位
- cache总容量=16行x(32x8 + 20 + 1 + 1 + 1)=16x279=4464=558字节
- cache的有效位用来指出所在cache行中的信息是否有效

- 4.19 某计算机采用页式虚拟存储管理方式，按字节编址。CPU进行存储访问的过程如图4.57所示。请回答下列问题：
- (1) 主存的物理地址占多少位？
- (2) TLB采用什么映射方式？TLB是用SRAM还是DRAM实现？
- (3) cache采用什么映射方式？若cache采用LRU替换算法和写回策略，则cache每行中除数据（Data）、tag和有效位外，还应有哪些附加位？Cache总容量是多少？Cache中有效位的作用是什么？
- (4) 若CPU给出的虚拟地址为0008 C040H，则对应的物理地址是多少？是否在cache中命中？说明理由。若CPU给出的虚拟地址是0007 C260H，则该地址所在主存块映射到的cache组号是多少？



解（续）：

— (4)

- 虚拟地址为0008 C040H= 0008C 040
- 根据0008CH查TLB，得到实页号=0040H，并且有效位=1
- 因此，物理地址=0040 040H
- 物理地址=标记（20位）+组索引（3位）+块内偏移（5位）=0040 040H=0000 0000 0100 0000 0000 010 00000
- 组索引=010=2，标记=00400H
- 查二路组相联映射cache的第2组，有两个标记（00400H、01600H），有一个比较相等，但是其有效位=0，不命中
- 虚拟地址为0007 C260H= 0007C 260=VPN（20位）+ VPO（12位）
- 主存地址=PPN（16位）+ PPO（12位）= PPN（16位）+ 260H = xxxx xxxx xxxx xxxx 0010 011 00000
- 因此，组索引=011=3，该地址所在主存块映射到的cache组号是3

Thanks