

计算机组成原理课程作业

——第四次

黄勖 22920212204392



4.1

解释下列名词。

1. 存取时间：存取时间指的是从发出存储器访问请求到数据可供使用所需的时间。存储器访问的速度是计算机性能的重要指标之一，存取时间越短，计算机性能越高。
2. 存取周期：存取周期是存储器中一次读写操作所需的时间，通常是存储器操作的最小时间单位。
3. 存储器带宽：存储器带宽是指在一定时间内从存储器中读取或写入数据的速率，通常以字节/秒或者位/秒来表示。
4. 存储单元：存储单元是存储器中最小的可寻址存储位置，通常是一个字节或者一个字（多个字节构成）。
5. 边界对齐的数据存放：边界对齐是指将数据存储在地地址为某个值的倍数的存储单元中。边界对齐的数据存放可以提高存取数据的效率。
6. 大端存储和小端存储：大端存储和小端存储是两种不同的数据存储方式。在大端存储中，数据的高位存储在低地址处，而低位存储在高地址处；而在小端存储中，数据的低位存储在低地址处，而高位存储在高地址处。
7. 静态存储器和动态存储器：静态存储器和动态存储器是两种不同的存储器类型。静态存储器通常由触发器构成，不需要刷新，速度较快，但成本较高；而动态存储器通常由电容构成，需要定期刷新，速度较慢，但成本较低。
8. 刷新和刷新周期：刷新是指动态存储器中定期将存储的数据重新写回电容中的操作，以防止数据丢失。刷新周期是指刷新操作的时间间隔。

4.1

解释下列名词。

9. 位扩展和字扩展：位扩展是指增加存储的字长。一块实际的存储芯片,其每个单元的位数往往与实际内存单元字长并不相等。存储芯片可以是1位、4位或8位,而计算机内存一般是按字节来进行组织的,单个存储芯片字长(位数)不能满足要求,这时就需要进行位扩展,以满足内存单元字长的要求。

字扩展是对存储器容量的扩展(或存储空间的扩展)。此时存储器芯片上每个存储单元的字长已满足要求(如字长已为8位),而只是存储单元的个数不够,需要的是存储单元的数量,这就是字扩展,即用多片字长为8位的存储芯片构成所需要的存储空间。

在构成一个实际的存储器时,往往需同时进行位扩展和字扩展才能满足存储容量的要求,这叫字位扩展。

10. 多体交叉存储器：多体交叉存储器是一种存储器结构,将存储器划分为多个体,每个体可以独立访问。通过交叉访问多个体,可以提高存储器的带宽和访问速度。

11. 高速缓冲存储器：高速缓冲存储器 (Cache) 是一种中间层存储器,通常位于CPU和主存之间,用于加速对主存的访问。Cache可以利用时间局部性和空间局部性原理,缓存最近经常访问的数据,以提高访问速度。

12. 双端口存储器：双端口存储器是一种具有两个独立访问端口的存储器,可以同时进行读写操作。双端口存储器通常用于需要高速读写操作的场景,如视频处理、网络交换等。

13. 相联存储器：相联存储器 (Associative Memory) 是一种根据内容而非地址来访问数据的存储器。相联存储器可以实现快速的数据查找,常用于高速缓存、模式识别等领域。

14. 时间局部性：时间局部性指的是程序在某个时间点访问某个数据的概率较高。通过利用时间局部性,可以将最近访问的数据缓存在高速缓存中,以提高访问速度。

4.1

15.地址映射：地址映射是将程序中的逻辑地址映射为实际物理地址的过程。地址映射通常由操作系统完成。

16.直接相联映射：直接相联映射是一种地址映射方式，将每个逻辑地址映射到唯一的物理地址。直接相联映射的缓存结构简单，但冲突较多，容易导致缓存命中率较低。

17.全相联映射：全相联映射是一种地址映射方式，将所有逻辑地址映射到任意一个物理地址。全相联映射的缓存结构复杂，但冲突较少，命中率较高。

18.组相联映射：组相联映射是一种地址映射方式，将逻辑地址分为多个组，每个组中包含多个缓存块。组相联映射结合了直接相联映射和全相联映射的优点，既能减少冲突，又能降低复杂度。

19.命中率：命中率是指在访问缓存时，所访问到的数据在缓存中的比例。命中率越高，说明缓存的效果越好。

20.虚拟存储器：虚拟存储器是一种将磁盘空间作为内存扩展的技术，可以在物理内存不足时将部分数据存储在磁盘上，并通过分页机制将数据动态地映射到物理内存中，从而扩大程序可以使用的内存空间。

21.页框号：页框号是虚拟存储器中的一种地址格式，用于表示页表中对应的物理页框号。

22.页表(慢表)：页表是虚拟存储器中的一种数据结构，用于存储逻辑地址和物理地址的映射关系。由于页表需要频繁访问，通常存储在内存中，因此也称为“慢表”。

23.页表项：页表项是页表中的一个条目，用于记录逻辑页号和物理页框号之间的映射关系，以及一些标志位等信息。

4.1

24.TLB(快表): TLB是Translation Lookaside Buffer的缩写, 也称为快表, 是一种高速缓存, 用于加速虚拟地址到物理地址的转换过程。TLB通常存储最近访问的页表项, 可以大大提高地址转换速度。

25.LRU算法: LRU算法是Least Recently Used的缩写, 是一种缓存置换算法, 用于确定哪些数据应该被替换。LRU算法根据数据最近的使用时间来判断哪些数据是最久未使用的, 从而选择替换这些数据。

26.LFU算法: LFU算法是Least Frequently Used的缩写, 也是一种缓存置换算法。LFU算法根据数据被访问的频率来判断哪些数据使用次数最少, 从而选择替换这些数据。

27.cache一致性: cache一致性是指在多个缓存中存储同一数据时, 保证缓存中的数据是相同的, 从而避免数据不一致的情况。实现cache一致性的方式包括MSI协议、MESI协议、MOESI协议等。

28.写回法: 写回法是一种缓存写入策略, 即在缓存中修改数据时, 只在缓存中修改, 并不立即将数据写回到内存中, 而是在缓存块被替换或程序退出时, 再将缓存中的数据写回到内存中。

29.写穿法: 写穿法是一种缓存写入策略, 即在缓存中修改数据时, 同时将数据写回到内存中, 从而保证内存中和缓存中的数据一致性。相比写回法, 写穿法的缺点是会增加总线流量和内存访问时间, 但是可以保证数据一致性。

(1) [2010]下列有关RAM和ROM的叙述中，正确的是 (A)

I. RAM是易失性存储器，ROM是非易失性存储器

II. RAM和ROM都采用随机存取方式进行信息访问

III. RAM和ROM都可用作cache

IV. RAM和ROM都需要进行刷新

A. 仅I和II

B. 仅II和III

C. 仅I、II和IV

D. 仅II、III和IV

III. SRAM用作cache。 IV. 只有RAM (DRAM) 需要刷新。

4.2

插入一段文本插入

(2) [2014]某容量为256MB的存储器由若干4M x 8位的DRAM芯片构成，该DRAM芯片的地址引脚和数据引脚总数是 (A)

A.19 B.22 C.30 D.36

256/4=64，需要64个DRAM芯片。每个芯片的地址引线应为22根（因为4M=2²²B），但DRAM芯片有地址线复用技术（行列复用），仅需一半的引脚数即够用，所以芯片的地址引脚为11个。芯片的数据引脚为8个（因为是8位的芯片）。所以该芯片的地址引脚和数据引脚总数为11+8=19

4.2

插入一段文本插入

(3) [2009]某计算机主存容量为64KB,其中ROM区为4KB,其余为RAM区,按字节编址。现要用2KB x8位的ROM芯片和4KB x4位的RAM芯片来设计该存储器,则需要上述规格的ROM芯片数和RAM芯片数分别是 (D)

A.1、 15

B.2、 15

C.1、 30

D. 2、 30

ROM区大小: 4KB, ROM芯片大小: 2KB, 无需位扩展, 所需ROM芯片数量:
 $4/2=2$

RAM区大小: 60KB, RAM芯片大小: 4KB, 按字节编址, 一个字节即8位是一个单位, 所以要对RAM芯片进行位扩展, 因为芯片是4位, 所以要两个拼为一组使用, 则扩展后的一组芯片是4K×8位, 所需RAM芯片数量:

$60/4 \times 2=30$

4.2

插入一段文本插入

(4) [2010]假定用若干个2KB x 4位的芯片组成一个8KB x 8位的存储器，则地址0B1FH所在芯片的最小地址是（ D ）

A.0000H B.0600H C.0700H D.0800H

使用译码器将4个11位（ $2K=2^{11}B$ ）的芯片地址转换成13位的存储器地址，则地址0B1F = 0000 1011 0001 1111，最高位的两位代表的即使选择的芯片地址，则选择的芯片地址的最小地址应为0000 1000 0000 0000，即0800H。

4.2

插入一段文本插入

4.2

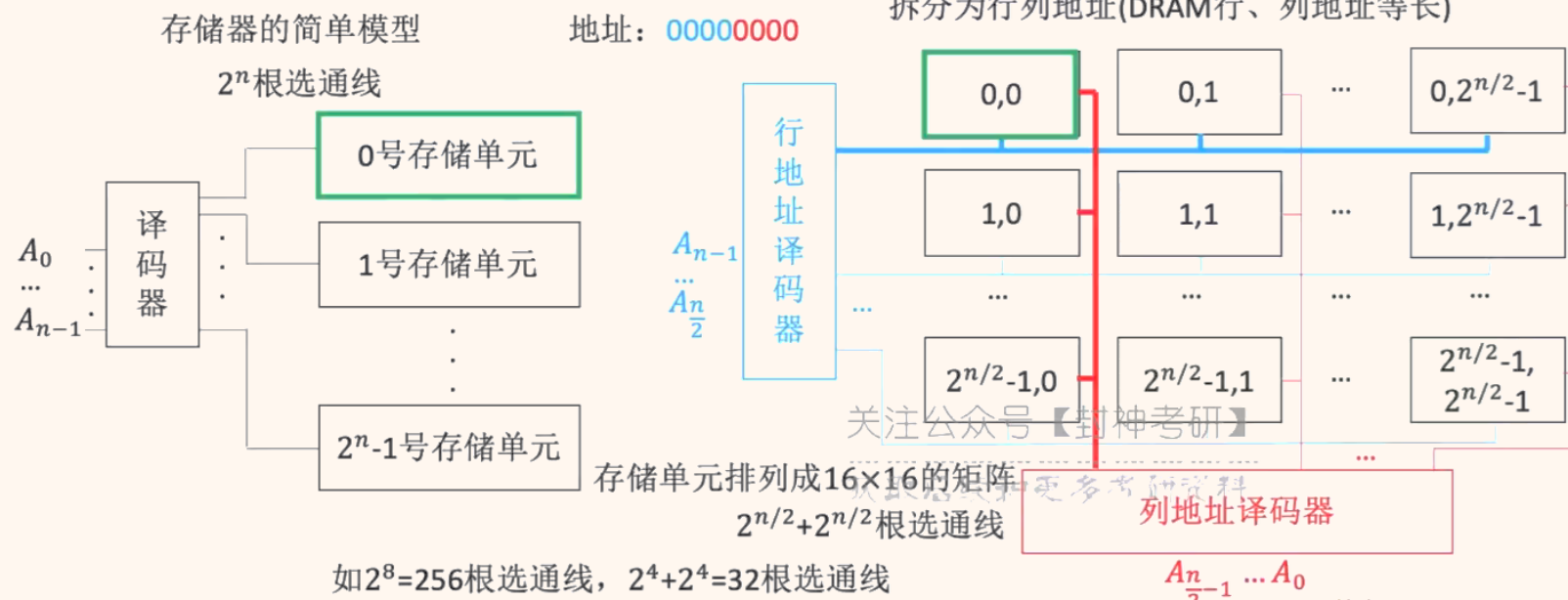
(5) [2018]假定DRAM芯片中存储阵列的行数为 r 、列数为 c ,对于一个2KB x 1位的DRAM芯片,为保证其地址引脚数最少,并尽量减少刷新开销,则 r 、 c 的取值分别是 (C)

A.2048、1 B.64、32 C. 32、64 D.1、2048

复用时,芯片需要11条地址线,一个取5,一个取6时管脚数最小(行列数相近),又已知DRAM行数越多刷新开销越大(以行为单位,每次刷新1行存储单元),所以为减少刷新开销,行地址线取5,列地址线取6,则行数 $r=2^5=32$,列数 $c=2^6=64$ 。

DRAM的刷新

1. 多久需要刷新一次? 刷新周期: 一般为2ms
2. 每次刷新多少存储单元? 以行为单位, 每次刷新一行存储单元
——为什么要用行列地址? 减少选通线的数量



4.2

(5) [2018]假定DRAM芯片中存储阵列的行数为 r 、列数为 c ,对于一个2KB x 1位的DRAM芯片,为保证其地址引脚数最少,并尽量减少刷新开销,则 r 、 c 的取值分别是 (C)

A.2048、1 B.64、32 C. 32、64 D.1、2048

复用时,芯片需要11条地址线,一个取5,一个取6时管脚数最小(行列数相近),又已知DRAM行数越多刷新开销越大(以行为单位,每次刷新1行存储单元),所以为减少刷新开销,行地址线取5,列地址线取6,则行数 $r=2^5=32$,列数 $c=2^6=64$ 。

3. 如何刷新? 有硬件支持,读出一行的信息后重新写入,占用1个读/写周期

4. 在什么时刻刷新?

存取周期

假设DRAM内部结构排列成 128×128 的形式,读/写周期0.5us

2ms共 $2ms/0.5us = 4000$ 个周期

思路一: 每次读写完都刷新一行

→系统的存取周期变为1us

前0.5us时间用于正常读写

后0.5us时间用于刷新某行



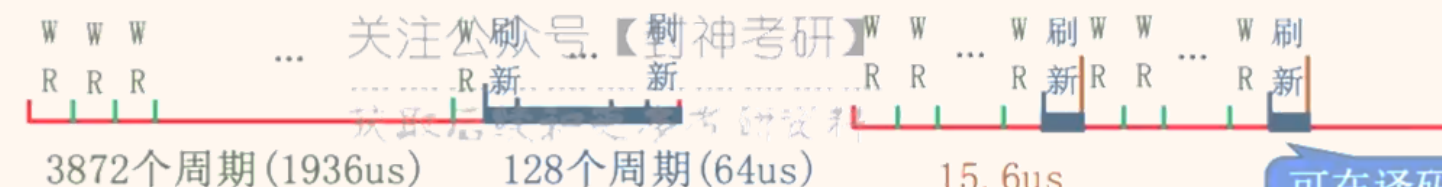
分散刷新

思路二: 2ms内集中安排时间全部刷新

→系统的存取周期还是0.5us

有一段时间专门用于刷新,

无法访问存储器,称为访存“死区”



集中刷新

思路三: 2ms内每行刷新1次即可

→2ms内需要产生128次刷新请求

每隔 $2ms/128 = 15.6us$ 一次

每15.6us内有0.5us的“死时间”



异步刷新

可在译码阶段刷新

(6) [2019]假定一台计算机采用3通道存储器总线，配套的内存条型号为DDR3-1333,即内存条所接插的存储器总线的工作频率为1333MHz、总线宽度为64位，则存储器总线的总带宽大约是 (B)

A.10.66GB/s B.32GB/s C.64GB/s D.96GB/s

总带宽=单位时间传输的数据长度/单位时间

$$=3*64/T(\text{位/s})$$

$$=3*64*f(\text{位/s})$$

$$=3*64*1333*2^{20}(\text{位/s})$$

$$=3*64*1333*2^{20}/2^{33}(\text{GB/s})=31.242\text{GB/s}。$$

4.2

插入一段文本插入

(7) [2015]某计算机使用4体交叉编址存储器，假定在存储器总线上出现的主存地址(十进制)序列为8005、8006、8007、8008、8001、8002、8003、8004、8000,则可能发生访存冲突的地址对是 (D)

A.8004 和8008 B.8002 和8007 C.8001 和8008 D.8000 和8004

主存地址模4后对应的体号分别是：1,2,3,4,1,2,3,4,4。8004和8000都是存取的四号存储器且被连续访问，可能导致8004存储还未完成而又存取8000地址，因此可能发生缓存冲突。

4.2

插入一段文本插入

(8) [2015]下列存储器中，在工作期间需要周期性刷新的是 (B)

A. SRAM B. SDRAM C. ROM D. FLASH

SRAM以双稳态为存储单元，数据不易丢失，不需要周期性的刷新。

DRAM的数据以电荷的形式保存在电容中，需要以刷新的方式，不断为电容充放电，需要周期性刷新。

(9) [2011]下列各类存储器中，不采用随机存取方式的是 (B)

A. EPROM B. CDROM C. DRAM D. SRAM

光盘采用顺序存储。

4.2

(10) [2012]下列关于闪存(Flash Memory)的叙述中, 错误的是 (A)

- A.信息可读可写, 并且读、写速度一样快
- B.存储元由MOS管组成, 是一种半导体存储器
- C.掉电后信息不丢失, 是一种非易失性存储器
- D.采用随机访问方式, 可替代计算机外部存储器

闪存的写操作必须在空白区域进行, 如果目标区域已经有数据, 必须先擦除后写入, 而读操作不必如此, 所以闪存的读速度比写速度快。

(11) [2017]下列关于数组a的访问局部性的描述中, 正确的是 (A)

- A.时间局部性和空间局部性皆有
- B.无时间局部性, 有空间局部性
- C.有时间局部性, 无空间局部性
- D.时间局部性和空间局部性皆无

数组a按顺序储存, 在对数组a进行遍历时, 是在连续按顺序访问地址空间, 具有空间局部性。

数组中的数据在遍历循环中会被多次访问, 则也会具有时间局部性。

(12) [2009] 某计算机的cache共有16块, 采用二路组相联映射方式(即每组2块)。每个主存块大小为32B, 按字节编址。主存129号单元所在主存块应装入的cache组号是 (C)

- A. 0
- B. 1
- C. 4
- D. 6

129号单元为129B的位置, 则在主存块的 $129/32=4\cdots 1$, 也就是第五块主存块, 块号为4, cache有 $16/2=8$ 组, 组号=主存块号 mod 组数, $4\%8=4$, 则组号=4

(13) [2012]假设某计算机按字编址，cache有4行，cache和主存之间交换的块大小为1个字。若cache的内容初始为空，采用二路组相联映射方式和LRU替换策略。访问的主存地址依次为0、4、8、2、0、6、8、6、4、8时，命中cache的次数是 (A/C)

A.1 B.2 C.3 D.4

A答案：

唐朔飞老师《计算机组成原理》：0.2.4.6.8映射到第0组cache，1.3.5.7.9映射到1组cache
cache共2组，每组2行。LRU算法根据数据最近的使用时间来判断哪些数据是最久未使用的，从而选择替换这些数据。

	0	4	8	2	0	6	8	6	4	8
0 组 1 行	0	0	8	8	0	0	8	8	4	8
0 组 2 行		4	4	2	2	6	6	6	6	4
1 组 0 行										
1 组 1 行										
	不命中	不命中	替换	替换	替换	替换	替换	命中	替换	替换

(13) [2012]假设某计算机按字编址，cache有4行，cache和主存之间交换的块大小为1个字。若cache的内容初始为空，采用二路组相联映射方式和LRU替换策略。访问的主存地址依次为0、4、8、2、0、6、8、6、4、8时，命中cache的次数是 (A/C)

A.1 B.2 C.3 D.4

C答案：（本书参考答案使用这种分组方式）

蒋本珊《计算机组成原理》（第二版及以前）：主存地址为0~1、4~5、8~9可映射到第0组Cache中，主存地址为2~3、6~7可映射到第1组Cache中；

第三版后分组方式同A答案；

	0	4	8	2	0	6	8	6	4	8
0 组 1 行	0	0	8	8	8	8	8	8	8	8
0 组 2 行		4	4	4	0	0	0	0	4	4
1 组 0 行				2	2	2	2	2	2	2
1 组 1 行						6	6	6	6	6
	不命中	不命中	替换	不命中	替换	不命中	命中	命中	替换	命中

(14) [2015] 假定主存地址为32位，按字节编址，主存和cache之间采用直接相联映射方式，主存块大小为4个字，每个字32位，采用写回的方式，则能存放4K字数据的cache的总容量至少是 (C) 位。

A.146K

B.147K

C.148K

D.158K

cache块个数=4K字/4字 (cache块大小=主存块大小) =1K

每个主存块大小为4*32=128位=16B=2^4B，则块内偏移地址为4位

块索引为10位 (1K=2^10B)。所以标识位为32-4-10=18位

表示如下：

主存块号		块内地址
28 位		4 位
18 位标记	10 位行号	4 位块内地址

写回法包含1位有效位和1位脏位，所以cache的总容量为4K*32+1K* (18+1+1) =148K

有效位 标记(19位)

1
0
0
0
0
0
0
0

0...01

Cache

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Cache: 8×64B = 512B

CPU 访问主存地址

0...01000 001110 :

①根据主存块号的后3位确定Cache行
②若主存块号的前19位与Cache标记匹配且有效位=1，则Cache命中，访问块内地址为001110的单元。

③若未命中或有效位=0，则正常访问主存

块号

主存

每个主存块的地址范围

0	0...0000000000~ 0...0000111111
1	0...0001000000~ 0...0001111111
2	0...0010000000~ 0...0010111111
8	0...1000000000~ 0...1000111111
2 ²² -3	1...1101000000~ 1...1101111111
2 ²² -2	1...1110000000~ 1...1110111111
2 ²² -1	1...1111000000~ 1...1111111111

关注公众号【封神考研】

获取更多资料和更多考研资料

主存: 256MB

4.2

(15) [2014]采用指令cache与数据cache分离的主要目的是. (**D**)

A.降低cache的缺失损失

B.提高cache的命中率

C.降低CPU平均访存时间

D.减少指令流水线资源冲突

分离cache的主要原因是执行部件存取数据时，指令预取部件又要取指令又要取数据，两者会发生冲突，采用独立的就可以解决这个问题，则应选择D。

4.2

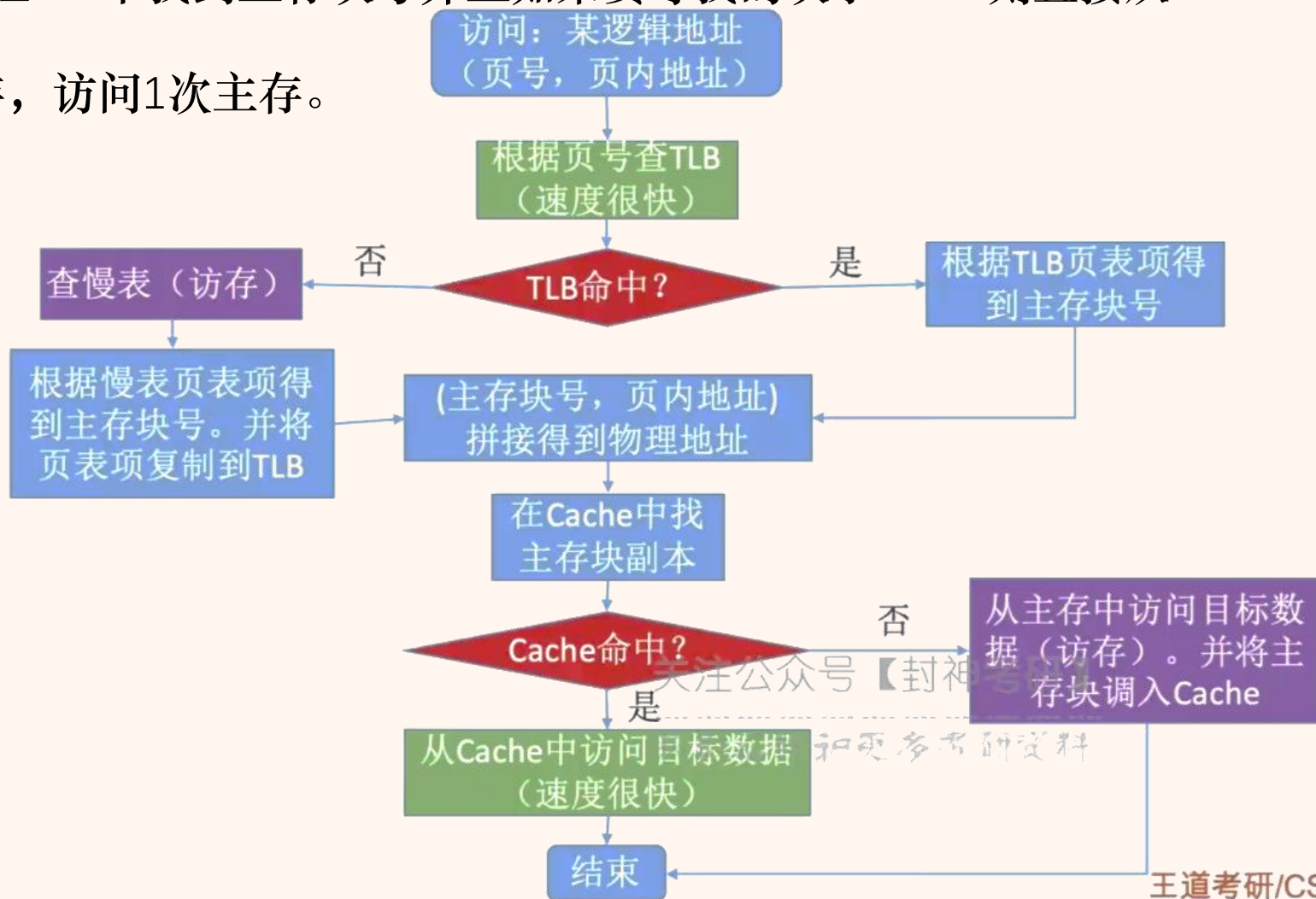
插入一段文本插入

(16) [2015] 假定编译器将赋值语句“ $x=x+3;$ ”转换为指令“add xaddr,3”，其中，xaddr 是x对应的存储单元地址。若执行该指令的计算机采用页式虚拟存储管理方式，并配有相应的TLB，且cache使用写穿的方式，则完成该指令功能需要访问主存的次数至少是 (B)

A. 0 B.1 C.2 D.3

取数步骤：可能不需要访问主存，直接在TLB中找到主存块号并且如果要寻找的块于cache则直接从cache中取出x的值。

写穿步骤：把数据同时写入cache和主存，访问1次主存。
则至少访问一次主存



(17) [2010] 下列命中组合情况中，一次访存过程中不可能发生的是 (**D**)

A. TLB未命中，cache 未命中，Page 未命中

B. TLB 未命中，cache 命中，Page 命中

C. TLB命中，cache未命中，Page命中

D. TLB 命中，cache 命中，Page 未命中

D: 页缺失说明页和数据不在主存中，TAB中一定没有对应页表项，cache也不可能命中，所以D不可能

4.2

(18) [2013]某计算机主存地址空间大小为256MB, 按字节编址。虚拟地址空间大小为4GB,采用页式存储管理方式, 页面大小为4KB, TLB (快表)采用全相联映射, 有4个页表项, 内容如表4.12所示。

表 4.12 4 个页表项的内容

有效位	标记	页框号	...
0	FF180H	0002H	...
1	3FFF1H	0035H	...
0	02FF3H	0351H	...
1	03FFFH	0153H	...

则对虚拟地址: 03FFF180H进行虚实地址转换的结果是 (A)

A.0153180H

B.0035180H

C. TLB 缺失

D.缺页

虚拟地址空间=4GB= 2^{32} B, 所以虚地址有32位。页面大小=4KB= 2^{12} B, 所以页内地址有12位, 所以虚页号=32-12=20位

主存地址空间=256MB= 2^{28} B, 所以实地址有28位, 页内地址与虚地址相同=12位, 所以页框号=28-12=16位。

转换: 03FFF180H, 其中页号为前20位: 03FFFH, 页内地址为后12位: 180H。对应的页框号为0153H。所以物理地址为0153180H。

4.2

(19) [2019]下列关于缺页处理的叙述中, 错误的是 (D)

A.缺页是在地址转换时CPU检测到的一种异常

B.缺页处理由操作系统提供的缺页处理程序完成

C.缺页处理程序根据页故障地址从外存读入所缺失的页

D.缺页处理完成后执行发生缺页的指令的下一条指令

缺页处理完成后将刷新缺页异常状态并回到发生缺页的指令 (原进程) 继续执行

4.2

插入一段文本插入

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：计算机系统中采用层次化存储体系结构的目的是为了提高存储器的访问速度和效率。在层次化存储体系结构中，存储器按照速度和容量划分为多个层次，速度较快的存储器位于较低层次，容量较大的存储器位于较高层次，计算机可以根据需要快速访问低层次存储器中的数据，而只在需要时才访问高层次存储器中的数据。这种层次化的存储器设计可以有效地提高计算机的性能和响应速度。

(2)为什么在存储器芯片中设置片选输入端？

答：在存储器芯片中设置片选输入端是为了选择要访问的存储器芯片。当计算机需要访问存储器系统中的某个特定芯片时，它会向该芯片发送一个片选信号，告诉芯片它被选中，其他芯片则被忽略。这种方式可以有效地减少存储器系统中的冲突和干扰，提高存储器的访问速度和稳定性。

(3)动态MOS存储器为什么要刷新？如何刷新？

答：动态MOS存储器为了保持存储器中的数据不失效，需要定期刷新。刷新的原因是动态MOS存储器中的电容器存储单元会逐渐失去电荷，导致存储的数据丢失。为了防止数据丢失，存储器需要定期进行刷新，将电容器中的电荷重新充满。刷新过程可以通过周期性地读取并重写存储单元中的数据来完成，这样可以使电容器中的电荷保持在足够高的水平，从而保持存储器中的数据有效。刷新的频率取决于存储器的设计和使用情况，一般在几十毫秒到几秒之间。

(4)试述多体交叉存储器的设计思想和实现方法。

答：多体交叉存储器是一种高速缓存存储器的设计方案。其设计思想是将存储器划分为多个体，每个体中包含多个存储块，每个存储块包含多个存储单元。不同的体之间通过交叉互连的方式进行连接，以实现高速缓存存储器对主存储器的数据访问。

多体交叉存储器的实现方法是通过将主存储器的地址分成多个部分，其中一部分用于选择体，另一部分用于选择存储块，最后一部分用于选择存储单元。这样，访问存储器时，可以先选择所需的体和存储块，然后在该存储块中选择所需的存储单元进行读写操作。

(5)为什么说cache对程序员是透明的?

答: Cache对程序员是透明的, 是因为程序员不需要直接管理Cache, Cache的读写操作是由计算机硬件自动完成的。当程序需要访问存储器时, 计算机首先检查Cache中是否已经包含了所需的数据, 如果已经存在, 则直接从Cache中读取数据, 不需要访问主存储器; 如果不存在, 则需要从主存储器中读取数据, 并将数据存储到Cache中。这个过程对程序员来说是透明的, 他们不需要了解Cache的具体细节, 也不需要自己管理Cache。

(6)直接相联映射方式下为什么不需要使用替换算法?

答: 直接相联映射方式下不需要使用替换算法, 是因为每个主存储器块只能映射到Cache中的一个特定的Cache块中。因此, 在Cache中如果已经存在该Cache块, 则直接使用该Cache块中的数据; 如果不存在, 则需要将主存储器块中的数据存储在空的Cache块中。由于Cache块数有限, 当Cache中所有的块都被使用时, 再次需要存储新的主存储器块时, 就需要替换掉一个已经存在的Cache块。但是, 在直接相联映射方式下, 替换掉的Cache块只能是对应的主存储器块, 因此不需要使用替换算法进行选择。

(7)为什么要考虑cache的一致性?

答: Cache的一致性是指多个处理器或者核心共享同一块主存储器区域时, 各自的Cache中的数据是否保持一致。由于各个处理器的Cache是独立的, 所以在多处理器或者多核心系统中, 当一个处理器或者核心修改了共享数据时, 其他处理器或者核心中的Cache中的数据就会变得过期。因此, 为了保证数据的一致性, 需要在处理器或者核心之间进行一定的协调和同步, 以确保各个Cache中的数据保持一致。

(8)替换算法有哪几种?它们各有何优缺点?

答: 替换算法常见的有以下几种:

1. 随机替换算法: 随机选择一个块进行替换。优点是实现简单, 缺点是不能充分利用Cache的局部性原理, 替换效果不理想。
2. 先进先出替换算法 (FIFO): 按照块进入Cache的时间顺序进行替换。优点是实现简单, 缺点是无法考虑块的重要性, 不能充分利用块的访问频率信息。
3. LRU算法: LRU算法是Least Recently Used的缩写, 是一种缓存置换算法, 用于确定哪些数据应该被替换。LRU算法根据数据最近的使用时间来判断哪些数据是最久未使用的, 从而选择替换这些数据。缺点是需要额外的计数器来统计块的使用时间, 实现复杂。
4. LFU算法: LFU算法是Least Frequently Used的缩写, 也是一种缓存置换算法。LFU算法根据数据被访问的频率来判断哪些数据使用次数最少, 从而选择替换这些数据。缺点是需要额外的计数器来统计块的使用频率, 实现复杂。

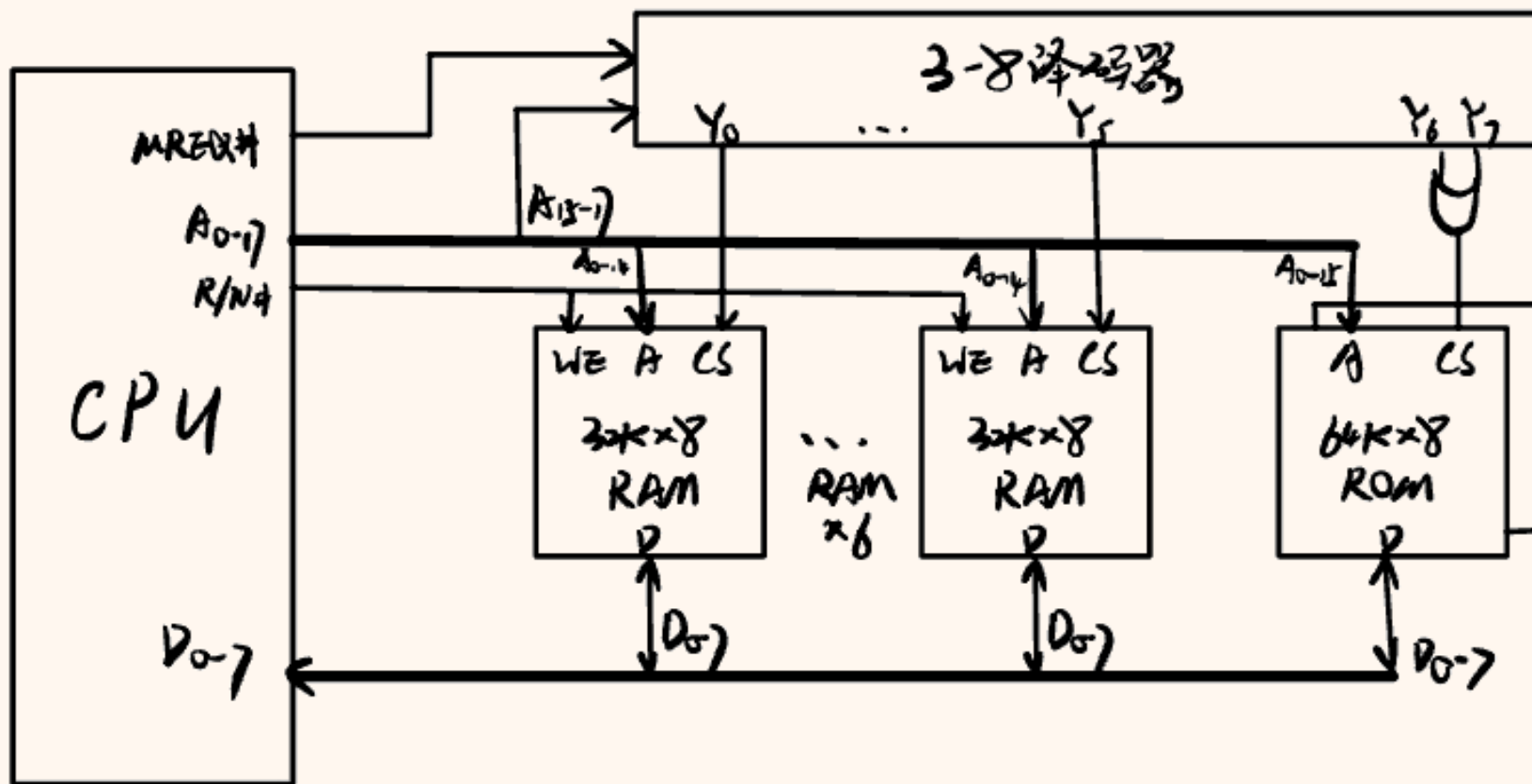
不同的替换算法各有优缺点, 需要根据具体应用场景进行选择。例如, 如果应用程序的访问模式比较随机, 可以选择随机替换算法; 如果应用程序的访问模式具有较好的局部性, 可以选择最少使用替换算法。

4.6用32K x 8位RAM芯片和64K x 4位ROM芯片设计256K x 8位存储器。其中，从30000H到3FFFFH的地址空间为只读存储区，其他为可读、可写存储区。完成存储器与CPU的连接。

30000H~3FFFFH一共有65536=64K个地址，ROM区需要2个进行位扩展，组成64K×8位的存储器，剩余RAM区需要6个进行字扩展，组成192K×8位的RAM。

设计如下：

4.6



4.6

4.8用64K x 1位的DRAM芯片构成1M x 8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0.5 μ s,若采用集中刷新方式，则存储器刷新一遍最少要用多少个读写周期? CPU的“死”时间为多少?

答：DRAM芯片存储阵列为64K=2¹⁶=2⁸×2⁸=256x256，所以共有256行

采用异步刷新：刷新周期为2ms，所以产生刷新信号的时间间隔
=2ms/256=7.8125 μ s。

采用集中刷新：存储器刷新一遍至少需要 256个读写周期，读写周期为0.5微秒，
所以CPU 的死时间256x0.5 μ s=128 μ s。

4.8

插入一段文本插入

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)。

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	9ABEFC0D		

将主存中这些数据装入cache 后，cache 各块中的数据内容及相应的标志是什么？

(1)全相联映射；

32位主存地址 = 30位标志+2位块内地址

标志取主存块号，每块 $32/8=4B(=2^2)$ 即去掉后两位)

8H=1000→10=2H

010004H=0001 0000 0000 0000 0100→0000 0100 0000 0000 0001=004001H

1FFFC6H= 0001 1111 1111 1111 1100→0111 1111 1111 1111=7FFFH

FFFFF8H=1111 1111 1111 1111 1111 1000→0011 1111 1111 1111 1111 1110=3FFFFEH

cache 行	标志	数据
0	000000	87568536
1	000002	87792301
2	004001	9ABEFC0D
3	007FFF	4FFFFC68
4	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)。

将主存中这些数据装入cache后，cache各块中的数据内容及相应的标志是什么？

(1) 全相联映射；

“全相联映射”如何访存？

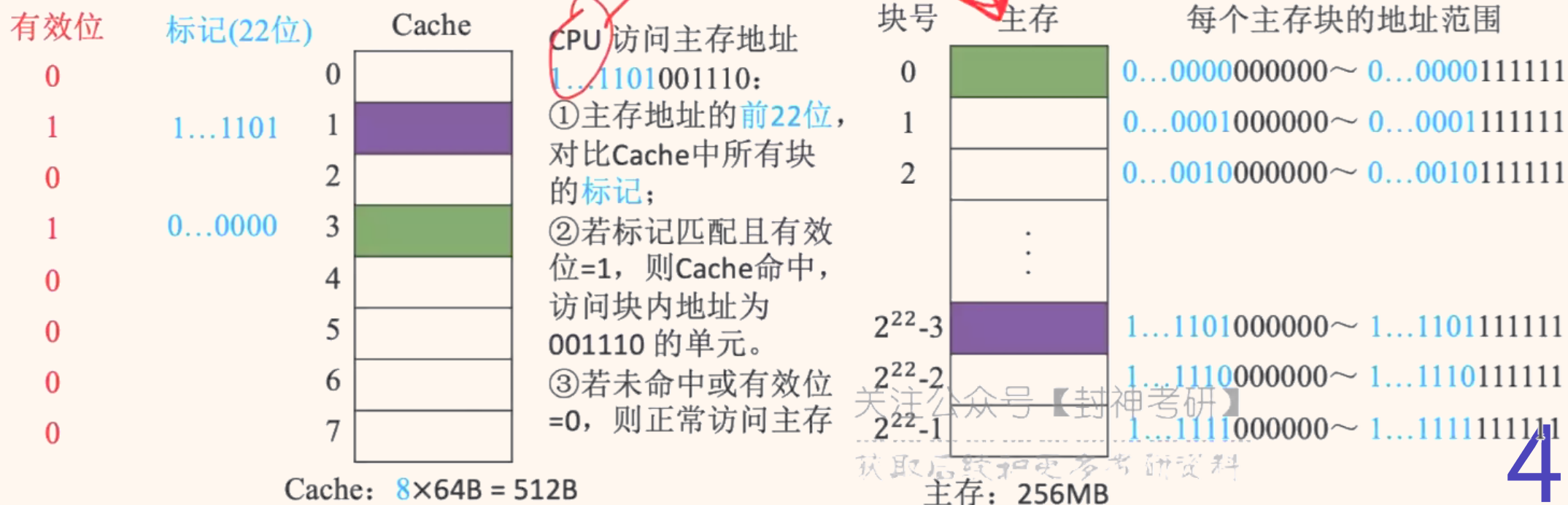
中国大学MOO

假设某个计算机的主存地址空间大小为256MB，按字节编址，其数据Cache有8个Cache行，行长为64B。

即Cache块，与主存块的大小相等

$256\text{M}=2^{28}$ 主存的地址共28位：

主存块号	块内地址
22位	6位



4.11

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)。

(2) 直接相联映射;

“直接映射”如何访存

假设某个计算机的主存地址空间大小为256MB，按字节编址，其数据Cache有8个Cache行，行长为64B。

直接映射，主存块在Cache中的位置=主存块号%Cache总块数

$256\text{M}=2^{28}$ 主存的地址共28位

主存块号		块内地址
22位		6位
19位 标记	3位 行号	6位块内 地址

Cache 共 2^3 行

每个主存块的地址范围

有效位	标记(19位)	Cache
1	0...01	0 000
0		1 001
0		2 010
0		3 011
0		4 100
0		5 101
0		6 110
0		7 111

Cache: $8 \times 64\text{B} = 512\text{B}$

CPU 访问主存地址
0...01000 001110 :

①根据主存块号的后3位确定Cache行

②若主存块号的前19位与Cache标记匹配且有效位=1，则

Cache命中，访问块内地址为001110的单元。

③若未命中或有效位=0，则正常访问主存

块号 主存

0	
1	
2	
8	
$2^{22}-3$	
$2^{22}-2$	
$2^{22}-1$	

关注公众号【封神考研】

获取更多考研资料

主存: 256MB

0...0000000000 ~ 0...0000111111
0...0001000000 ~ 0...0001111111
0...0010000000 ~ 0...0010111111
0...1000000000 ~ 0...1000111111
1...1101000000 ~ 1...1101111111
1...1110000000 ~ 1...1110111111
1...1111000000 ~ 1...1111111111

4.11

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)。

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	9ABEFC0D		

(2)直接相联映射;

Cache行号 2^{14} 个

即32位主存地址 = 8位标志+14位行号+2位块内地址

如: FFFFF8H=1111 1111 1111 1111 1111 1000

其余以此类推

cache 行	标志	数据
0000	00	87568536
0002	00	87792301
0001	01	9ABEFC0D
3FFF	01	4FFFFC68
3FFE	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)。

(3)四路组相联映射。

“组相联映射”如何访存

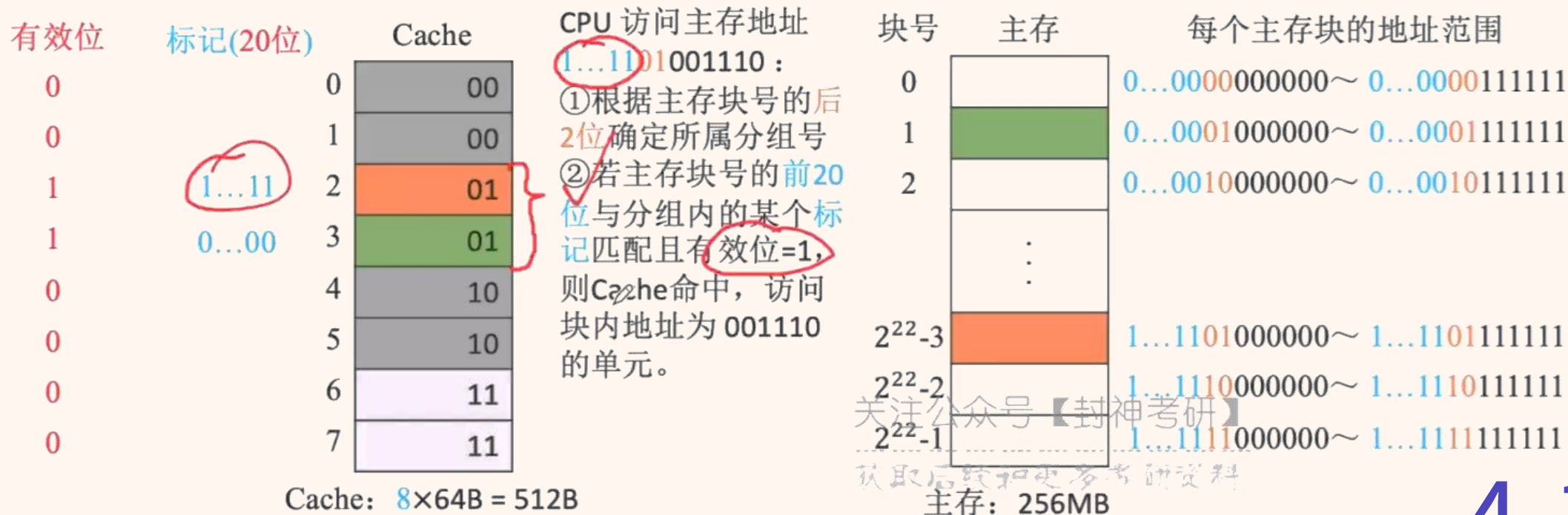
假设某个计算机的主存地址空间大小为256MB，按字节编址，其数据Cache有8个Cache行，行长为64B。

组相联映射，所属分组=主存块号%分组数

256M=2²⁸ 主存的地址共28位

主存块号		块内地址
22位		6位
20位 标记	2位 组号	6位块内 地址

Cache 分为2²²组



2路组相联映射——2块为一组，分四组

4.11

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)。

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	9ABEFC0D		

(3) 四路组相联映射。

Cache组数 $2^{14}/4 = 2^{12}$ 个

32位主存地址 = 10位标志 + 12位行号 + 2位块内地址

如：FFFFF8H = 1111 1111 11 (3FF) 11 1111 1111 10 (0FFE) 00

其余以此类推

cache 行	标志	数据
000	000	87568536
002	000	87792301
001	004	9ABEFC0D
0FFF	007	4FFFFC68
0FFE	3FF	01BF2460

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)cache每块8个字, 每字32位, 所以每块大小=32B, 共有 $16\text{KB}/32\text{B}=2^{14}\text{B}/2^5\text{B}=2^9=512$ 块。每组分为4块, 所以共有 $512/4=128$ 组。

从主存单元读出100个字, $100/8=12\cdots 4$, 所以主存分为13个块。第一次读取时需要将主存中每个块中的第一个字载入cache, 此时不命中。一共有13个主存块, 所以载入13次, 即不命中的次数为13。之后再读取主存中的其他字时都可在cache中找到即命中。所以cache命中率= $(100\times 10 - 13)/(100\times 10)=0.987$

(2) 设命中cache访问时间为t, 未命中cache (访问主存) 访问时间为10t

平均访问时间为 $0.987 \times t + (1 - 0.987) \times 10t = 1.117t$ 。

不采用cache (次次访问主存) 的平均访问时间为10t

所以使用cache的存储访问速度是不使用的 $10t/1.117t \approx 8.95$ 倍, 即存储访问速度提高了7.95倍

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。

```
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;
}
```

```
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;
}
```

答：(1) 程序 A 中对数组的访问（顺序访问）具有空间局部性，不具有时间局部性；程序 B 中对数组的访问不具有空间局部性，也不具有时间局部性。

(2) sum 因为在 for 循环中被多次访问，所以具有时间局部性。

(3) for 循环中的指令被反复执行，具有时间局部性。并且循环语句中的机器指令序列通常会顺序执行，因此也具有一定的空间局部性。

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?

答:(1)虚拟存储容量=2GB= 2^{31} B,所以虚拟地址共有31位。页大小=128KB= 2^{17} B,所以页内偏移地址为17位。则虚页号为 $31-17=14$ 位。故虚拟地址中高14位表示虚页号,低17位表示页内偏移地址。

TLB有四组,需要2位组索引 ($4=2^2$), 则TBL标记为 $14-2=12$ 位。

故虚页号中高 12 位表示 TLB 标记,低 2 位表示 TLB 组索引。

(2)主存容量=4KB= 2^{22} B,所以物理地址共有22位。页内偏移地址为17位,所以物理页号为 $22-17=5$ 位。故物理地址中高 5 位表示物理页号,低 17 位表示页内偏移地址。

(3)主存容量=4KB= 2^{22} B,所以主存地址有22位。

cache每块包含8个字,每字32位,所以每个cache块大小为 $32\text{B} = 2^5\text{B}$ 。

cache容量=16KB= 2^{14} B, cache块数量为 $2^{14}\text{B}/2^5\text{B}=2^9$ 。

$32\text{B}=2^5\text{B}$,块内地址为5位。

采用四路组相联映射,每四个为一组,所以cache组数为 $2^9/4=2^7$,所以组索引为7位。剩下的标记位= $22-5-7=10$ 位。

故主存地址划分: 标记 (10位) + 组索引 (7位) + 块内地址 (5位)



ABOUT

感谢观看

THANKS FOR WATCHING

---汇报人：黄勖