

第四章 存储器分层体系结构

2. 简单回答下列问题。(参考答案略)

- (1) 计算机内部为何要采用层次化存储体系结构? 层次化存储体系结构如何构成?
- (2) SRAM 芯片和 DRAM 芯片各有哪些特点? 各自用在哪些场合?
- (3) CPU 和主存之间有哪两种通信方式? SDRAM 芯片采用什么方式和 CPU 交换信息?
- (4) 为什么在 CPU 和主存之间引入 Cache 能提高 CPU 访存效率?
- (5) 为什么说 Cache 对程序员来说是透明的?
- (6) 什么是 Cache 映射的关联度? 关联度与命中率、命中时间的关系各是什么?
- (7) 为什么直接映射方式不需要考虑替换策略?
- (8) 为什么要考虑 Cache 的一致性问题? 读操作时是否要考虑 Cache 的一致性问题?
- (9) 什么是物理地址? 什么是逻辑地址? 地址转换由硬件还是软件实现? 为什么?
- (10) 什么是页表? 什么是快表 (TLB)?
- (11) 在存储器层次化结构中, “Cache-主存”、“主存-外存”这两个层次有哪些不同?

3. 已知某机主存容量为 64KB, 按字节编址。问:

- 1) 若用 1K×4 位的 SRAM 芯片构成该存储器, 需要多少片芯片?
- 2) 主存地址共多少位? 几位用于选片? 几位用于片内选址?
- 3) 画出该存储器的逻辑框图。

参考答案:

1) $64KB / 1K \times 4 \text{ 位} = 128 \text{ 片}$

2) 主存地址共 16 位, 6 位选片, 10 位片内选址。

4. 用 64K×1 位的 DRAM 芯片构成 256K×8 位的存储器, 要求:

- 1) 计算所需芯片数, 并画出该存储器的逻辑框图。
- 2) 若采用分散刷新方式, 每单元刷新间隔不超过 2ms, 则刷新信号周期是多少? 若采用集中刷新方式, 则存储器刷新一遍最少用多少读/写周期?

参考答案:

1) $256KB / 64K \times 1 \text{ 位} = 32 \text{ 片}$ 。

2) 采用分散刷新时, 刷新周期是存储周期的 2 倍;

因为 DRAM 芯片存储阵列为 $64K=256 \times 256$, 所以集中刷新时, 整个存储器刷新一遍需要 256 个

存储（读写）周期。

（异步刷新的刷新周期为 $2\text{ms}/256=7.8\mu\text{s}$ ）

6. 某计算机中已配有 0000H-7FFFH 的 ROM 区域，现在再用 8Kx4 位的 RAM 芯片形成 32Kx8 位的存储区域，CPU 地址总线为 A0-A15，数据总线为 D0-D7，控制信号为 R/W#（读/写）、MREQ#（访存）。要求说明地址译码方案，并画出 ROM 芯片、RAM 芯片与 CPU 之间的连接图。

参考答案：

CPU 地址线共 16 位，故存储器地址空间为：0000H – FFFFH，其中 8000H – FFFFH 为 RAM 区，共 $2^{15}=32\text{K}$ 个单元，容量为 32KB，故需 8Kx4 位的芯片数为 $32\text{KB}/8\text{Kx4 位}=4\text{x}2=8$ 片。

若 CPU 地址线为 24 位，ROM 区为 000000H – 007FFFH，则如何？

ROM 区大小为 32KB，总大小为 $16\text{MB}=2^{14}\text{KB}=512\text{x}32\text{KB}$ ，所以 RAM 区大小为 $511\text{x}32\text{KB}$

共需使用 RAM 芯片数为 $511\text{x}32\text{KB}/8\text{Kx4 位}=511\text{x}4\text{x}2$ 个芯片。

7. 假定一个存储器系统支持四体交叉存取，某程序执行过程中访问地址序列为 3, 9, 17, 2, 51, 37, 13, 4, 8, 41, 67, 10，则哪些地址访问会发生体冲突？

参考答案：

对于 4 体交叉访问的存储系统，每个存储模块的地址分布为：

Bank0: 0、4、8、12、16

Bank1: 1、5、9、13、17 ...37 ...41...

Bank2: 2、6、10、14、18

Bank3: 3、7、11、15、19...51...67

如果给定的访存地址在相邻的四次访问中出现在同一个 Bank 内，就会发生访存冲突。所以：

17 和 9、37 和 17、13 和 37、8 和 4、41 和 37 发生冲突。

10. 假定某机主存 1GB，按字节编址。Cache 的数据区（即不包括标记、有效位等存储区）有 64KB，块大小为 128 字节，采用直接映射和直写（Write-Through）方式。请问：

11) 主存地址如何划分？要求说明每个字段的含义、位数和在主存地址中的位置。

1) Cache 的总容量为多少位？

参考答案：

12) 主存：Cache 共 $64\text{KB}/128\text{B}=512$ 行。

主存地址为 30 位，其中，高 14 位为标志；中间 9 位为行索引（行号）；低 7 位为块内地址。

2) 考虑有效位、标志位而不考虑替换算法等所需的控制位，Cache总容量为：

$512 \times (128 \times 8 + 14 + 1) \text{ 位} = 519.5 \text{ K 位}$ 。

11. 假定某计算机的Cache共16行，开始为空，块大小为1个字，采用直接映射方式。CPU执行某程序时，依次访问以下地址序列：2，3，11，16，21，13，64，48，19，11，3，22，4，27，6和11。要求：

1) 说明每次访问是命中还是缺失，试计算访问上述地址序列的命中率。

2) 若 Cache 数据区容量不变，而块大小改为 4 个字，则上述地址序列的命中情况又如何？

参考答案

1) Cache：直接映射， $16 = 16 \text{ 块} \times 1 \text{ 字/块}$ ；主存：1 字/块，所以：主存块号=字号

映射公式为：Cache 块号 = 字号 mod 16

开始 Cache 为空，所以第一次都是 miss

2-2: miss, 3-3: miss, 11- 11: miss, 16- 0: miss, 21-5: miss, 13-13: miss, 64-0: miss、replace

48-0: miss、replace, 19-3: miss、replace, 11-11: hit, 3-3: miss、replace, 22-6: miss,

4-4: miss, 27-11: miss、replace, 6-6: miss、replace, 11-11: miss、replace

只有一次命中！

2) Cache：直接映射， $16 = 4 \text{ 块} \times 4 \text{ 字/块}$ ；主存：4 字/块，所以：主存块号=[字号/4]

映射公式为：Cache 块号 = 主存块号 mod 4

开始 Cache 为空，所以第一次都是 miss

2-0-0: miss; 3-0-0: hit; 11-2-2: miss, 16-4-0: miss, replace, 21-5-1、13-3-3: miss

64-16-0、48-12-0、19-4-0: miss, replace, 11-2-2: hit, 3-0-0: miss、replace,

22-5-1: hit, 4-1-1: miss、replace, 27-6-2: miss、replace, 6-1-1: hit, 11-2-2: miss、replace

命中 4 次。

块变大后，提高了空间局部性的利用率，从而使命中率提高！

12. 假定数组元素在主存按从左到右的下标顺序存放。试改变下列函数中循环的顺序，使得其数组元素的访问与排列顺序一致，并说明为什么修改后的程序比原来的程序执行时间短。

```
int sum_array ( int a[N][N][N])
```

```
{
```

```
    int i, j, k, sum=0;
```

```

for (i=0; i < N; i++)
    for (j=0; j < N; j++)
        for (k=0; k < N; k++)    sum+=a[k][i][j];

return sum;
}

```

参考答案：

```

int sum_array ( int a[N][N][N])
{
    int i, j, k, sum=0;

    for (k=0; k < N; k++)
        for (i=0; i < N; i++)
            for (j=0; j < N; j++)    sum+=a[k][i][j];

    return sum;
}

```

修改后程序的数组元素的访问与排列顺序一致，使得空间局部性比原程序好，故执行时间更短。

17. 假设某计算机有64MB主存，采用字节编址方式。其Cache数据区容量为4KB，采用4路组相联映射、LRU替换和写回（Write Back）策略，块大小为64B。请问：

- 1) 主存地址字段如何划分？要求说明每个字段的含义、位数和在主存地址中的位置。
- 2) 该Cache的总容量有多少位？
- 3) 若Cache初始为空，CPU依次从0号地址单元顺序访问到4344号单元，重复按此序列共访问16次。若Cache命中时间为1个时钟周期，缺失损失为10个时钟周期，则CPU访存的平均时间为多少时钟周期？

参考答案：

1) Cache: $4KB=2^{12}B=2^4$ 组 $\times 2^2$ 块/组 $\times 2^6$ 字节/块

主存地址划分为三个字段：

高 16 位为标志（tag）、中间 4 位为组号（Index）、最低 6 位为块内地址（Offset）。

即主存空间划分： $64MB=2^{26}B=2^{16}$ 组群 $\times 2^4$ 块/组群 $\times 2^6$ 字节/块

2) Cache 共有 64 行，每行中有：16 位标志、1 位有效位、1 位修改位、2 位 LRU 位和数据 64B。

故总容量为： $64 \times (16+1+1+2+64 \times 8)=34048$ 位。

3) 因为每块为 64B, $4345/64=67.89$ $67 \times 64=4288$

所以 0~4344 单元应该对应前 68 块, 也即处理器的访问过程是对前 68 块连续访问 16 次。

总访存次数: $16 \times 4345=69520$

Cache 共有 16 组, 每组 4 行, 替换情况如下图所示:

	第0行	第1行	第2行	第3行
0组	0/64/48	16/0/64	32/16	48/32
1组	1/65/49	17/1/65	33/17	49/33
2组	2/66/50	18/2/66	34/18	50/34
3组	3/67/51	19/3/67	35/19	51/35
4组	4	20	36	52
...
...
15组	15	31	47	63

LRU 算法: 第一次循环, 每一块只有第一次未命中, 其余都命中;

以后 15 次循环, 有 20 块的第一字未命中, 其余都命中.

所以, 命中率 p 为 $(69520-68-15 \times 20)/69520=99.47\%$

平均访存时间为: $p \times \text{Hit Time} + (1-p) \times \text{Miss Penalty}$

$$=1 \times p + 10 \times (1-p) = 1 \times 0.9947 + 10 \times 0.0053 = 1.0477 \text{ 个时钟周期}$$

18. 假定某处理器可通过软件对高速缓存设置不同的写策略, 那么, 在下列两种情况下, 应分别设置成什么写策略? 为什么?

1) 处理器主要运行包含大量存储器写操作的数据访问密集型应用。

2) 处理器运行程序的性质与 1) 相同, 但安全性要求高得多, 不允许有任何数据不一致的情况发生。

参考答案:

a) 采用 Write Back 策略较好, 可减少访存次数;

b) 采用 Write Through 策略较好, 能保证数据的一致性。

20. 提高关联度通常会降低失效率，但并不总是这样。请给出一个地址访问序列，使得采用LRU替换算法的2-路组相联映射Cache比具有同样大小的直接映射Cache的失效率更高。

参考答案：（略）

能举出很多例子。

2-way 组相联 Cache 的组数是直接映射 **Cache** 的项数的一半，所以，可以找到一个地址序列 **A、B、C**，使得：**A** 映射到一个 **Cache** 块，**B** 和 **C** 同时映射到另一个 **Cache** 块，并且 **A、B、C** 映射到同一个 **Cache** 组。这样，如果访存的地址序列为 **A、B、C、A、B、C、A、B、C...**，则：

对于直接映射 **Cache**，其命中情况为：

miss/miss/miss / hit/miss/miss / hit/miss/miss/... 命中率可达 **33.3%**。

对于 **LRU** 的组相联 **Cache**，因为 **A、B、C** 映射到同一个组，所以，每个地址处的数据刚调出 **Cache** 就又被访问到，每次都是 **miss**。命中率为 **0**。

例如：假定直接映射 **Cache** 为 **4 块 x1 字/块**

则同样大小的 **2-way 组相联 Cache** 为 **2 组 x2 块/组 x1 字/块**

当访问序列为：**0、2、4、0、2、4、0、2、4、...**（局部块大小为 **3**）时

对于直接映射，其对应的 **Cache** 块为：**0-2-0-0-2-0-0-2-0-...**

对于组相联映射，对应的 **Cache** 组为：**0-0-0-0-0-0-0-0-0-...**

当访问的局部块大于组的大小时，可能会发生“颠簸”现象：刚被替换出去的数据又被访问，导致失效率为 **100%**！

22. 假定某处理器带有一个数据区容量为256B的Cache，其块大小为32B。以下C语言程序段运行在该处理器上，sizeof(int)=4，编译器将变量i, j, c, stride都分配在通用寄存器中，因此，只要考虑数组元素的访存情况。若Cache采用直接映射方式，则当stride=256和stride=255时，失效率分别为多少？若Cache采用2-路组相联映射方式，则当stride=256和stride=255时，失效率又分别为多少？

```
int i, j, c, stride, a[512];  
.....  
for ( i = 0; i < 10000; i++ )  
    for ( j = 0; j < 512; j=j+stride )  
        c = a[j]+17;
```

参考答案：

已知 Cache 容量为 256B=8 块 x8 字/块 x4B/字，仅考虑数组访问情况。

- 1) 直接映射, stride=256: 访存顺序为 a[0], a[256], a[0], a[256], ... ,这两个元素一定映射到同一个 Cache 块中, 因此, 每次都发生冲突, 失效率为 100%
- 2) 直接映射, stride=255: 访存顺序为 a[0], a[255], a[510], a[0], a[255], a[510], ... ,这三个元素中后面两个元素映射到同一个 Cache 块中, 而 a[0] 不会发生冲突, 失效率为 67%
- 3) 2-way 组相联, stride=256: 访存顺序为 a[0], a[256], a[0], a[256], ... ,这两个元素虽然映射到同一个 Cache 组中, 但可以放在该组不同 Cache 块中, 失效率为 0%
- 4) 2-way 组相联, stride=255: 访存顺序为 a[0], a[255], a[510], a[0], a[255], a[510], ... ,这三个元素中后面两个元素虽映射到同一个 Cache 组中, 但可放在不同 Cache 块中, 而 a[0] 不会发生冲突, 故失效率为 0%

23. 假定一个虚拟存储系统的虚拟地址为40位, 物理地址为36位, 页大小为16KB。若页表中有效位、存储保护位、脏位、使用位共占4位, 磁盘地址不在页表中, 则该存储系统中每个进程的页表大小为多少? 如果按计算出来的实际大小构建页表, 则会出现什么问题?

参考答案:

每页大小有 16KB, 所以: 虚拟页数为: $2^{40}B/16KB=2^{(40-14)}=2^{26}$

物理页面和虚拟页面大小相等, 所以物理页号的位数为: $36-14=22$

页表项位数为: 有效位+保护位+脏位+使用位+物理页号位数= $22+4=26$ 位

为简化页表访问, 每项大小取 32 位。因此, 每个进程的页表大小为: $2^{26} \times 32b=256MB$

如果按实际计算出的页表大小构建页表, 则页表过大而导致页表无法一次装入。