

6. 使用地址中间位作为组索引、高位作为标签时，地址相邻的块会存放
在不同组中。若高位作组索引、中间位作标签，则地址相邻的块会被放
在缓存的同一个位置，发生冲突。当交替访问这些块时，缓存会发生抖动。

7. 让组索引和块内偏移的总位数与虚拟内存系统的页偏移位数相
同，使虚拟地址转换为物理地址的过程中，组索引不会改变。因此可以
直接用虚拟地址中页偏移的高位作为组索引访问缓存，而不必等到
物理地址后再访问缓存。

$$8. 1) 1 \times (1 - 3\%) + 110 \times 3\% = 4.27 \text{ (个) 周期}$$

$$2) 1 \times \frac{64 \text{ kB}}{1 \text{ GB}} + 110 \times \left(1 - \frac{64 \text{ kB}}{1 \text{ GB}}\right) \approx 109.99 \text{ 个周期}$$

3) 利用时间局部性和空间局部性，提前将主存中“很可能被访问”的数
据存入缓存中，可以提高缓存的命中率，减少存储系统的平均访问延时

4) 设命中率为 X

则使用缓存的平均访问延时： $X + 110 \times (1 - X)$

而禁用缓存为 105 周期

$$\text{令 } X + 110 \times (1 - X) < 105 \text{ 得: } X > 4.59\%$$

9. 编号	地址位数 Bit	块大小 KB	块大小 Byte	相联度	组数量	组索引位数 Bit	标签 位数 Bit	偏移 位数 Bit
						组索引位数 Bit	标签 位数 Bit	偏移 位数 Bit
1	32	4	64	2	32	5	21	6
2	32	4	64	8	8	3	23	6
3	32	4	64	全相联	1	0	26	6
4	32	16	64	1	256	8	18	6
5	32	16	128	2	64	6	19	7
6	32	64	64	4	256	8	18	6
7	32	64	64	16	64	6	20	6
8	32	64	128	16	32	5	20	7

10. 1) 系统 A: $\bar{t}_A = (1-P_1) \times 0.22 \text{ ns} + P_1 \times (0.22 \text{ ns} + 100 \text{ ns})$

系统 B: $\bar{t}_B = (1-P_2) \times 0.52 \text{ ns} + P_2 \times (0.52 \text{ ns} + 100 \text{ ns})$

令 $\bar{t}_A < \bar{t}_B$ 得: $P_1 - P_2 < 0.3\%$

2) $\bar{t}_A = (1-P_1) \times 0.22 \text{ ns} + P_1 \times (0.22k \text{ ns} + 0.22 \text{ ns})$

$\bar{t}_B = (1-P_2) \times 0.52 \text{ ns} + P_2 \times (0.52k \text{ ns} + 0.52 \text{ ns})$

令 $\bar{t}_A < \bar{t}_B$ 得: $P_1 < \frac{26}{11}P_2 + \frac{15}{11k}$

11. 1) 直接映射: 有 4 位作为组索引, 替换次数为 5

2) 2 路映射: 有 3 位作为组索引, 替换次数为 3

3) 4 路映射: 有 2 位作为组索引, 替换次数为 3

4) 8 路映射: 有 1 位作为组索引, 替换次数为 0

12. 1 个数的大小为 4 Byte, 1 个块中有 $16 \div 4 = 4$ 个数

一个缓存中有 $256 \div 16 = 16$ 个块

① 缓存 A: 每个块内的 4 个数据, 第一个缺失, 其余 3 个命中.
则缺失率为 $\frac{1}{4} = 25\%$

② 缓存 B: 当 $i=0$ 时, 缺失 24 次, 其余 $96 - 24 = 72$ 次命中

当 $i=1$ 到 $i=99$ 时, 对每个 i 的取值, 组中 16 个块中前 8 个块会被替换, 即会缺失 16 次, 其余 80 次命中.

缺失率为: $\frac{24 + 99 \times 16}{96 \times 100} = 16.75\%$

13.

```
for (int j=0; j<128; ++j) {
```

```
    for (int i=0; i<64; ++i) {
```

```
        A[j][i] = A[j][i] + 1;
```

```
}
```

14. 1) 有 $\frac{4KB}{32B} = 128$ 个块；一个数为 4 Byte，则一个块有 $\frac{32}{4} = 8$ 个数

优化前的缺失次数： $128 \times 64 = 8192 = 2^{13}$

优化后的缺失次数： $\frac{1}{8} \times 64 \times 128 = 1024 = 2^{10}$

2) 优化前的缺失次数： $\frac{1}{8} \times 128 \times 64 = 2^{10}$

优化后的缺失次数： $\frac{1}{8} \times 128 \times 64 = 2^{10}$

3) 优化前 $32\text{Byte} \times 128 = 4128\text{Byte}$

优化后 $32\text{Byte} \times 1 = 32\text{Byte}$

15.

	input 数组			output 数组				
	3 0	3 1	3 2	3 3	3 0	3 1	3 2	3 3

行0 miss miss hit hit miss miss miss miss

行1 hit miss miss hit miss miss miss miss

行2 hit miss hit hit miss miss miss miss

行3 hit miss miss hit miss miss miss miss

16. 1) 一个块中有 $\frac{16}{4} = 4$ 个数据，缓存有 $\frac{512}{16} = 32$ 个块

命中率为 75%

2) 不能；当一个块只有第一次被访问时会缺失，之后再次访问时均命中，命中率一直为 75%

3) 可以；增加块大小后，每块中数据增多，该块中的命中率增大，总的命中率增大。