

6. 解: 对地址相差一个块大小或不是很大的时候, 采用高地址位作索引就会导致映射到同一组中, 如果刚好程序^对这几个相邻数据来回访问, 则造成抖动。但如果采用中位作索引, 就会映射到不同组中。

7. 解: CPU向缓存发出的是虚拟地址, 因此到达缓存前需先翻译为物理地址, 为了减少命中时间, 可以采用索引是物理地址, 标签是物理地址的缓存, 所以在标签翻译为物理页号同时, 索引位^{块偏移}在缓存中索引组的位置, 所以此时索引位和块内偏移应等于页内偏移位数, 否则命中时间会增加。

8. 解 (1) $1 \times 97\% + 110 \times 3\% = 4.27$ 个周期

(2) 命中率为 $\frac{64KB}{1GB} = \frac{2^{16}}{2^{30}} = \frac{1}{2^{14}}$

延时为 $1 \times \frac{1}{2^{14}} + 110 \times (1 - \frac{1}{2^{14}}) \approx 109.99$ 个周期

(3) 当程序完全随机时, 这时缓存几乎每次访问都会使数据失效, 命中率很低, 无法利用局部性, 而当程序有局部性^部后, 可能很多访问数据会被多次, 缓存保存这些数据, 命中率提高, 访问延时降低。

(4) 设命中率为 A
 $1 \times A + 110 \times (1 - A) \leq 105$

$A \geq 4.59\%$ 时, L1有收益

9. 解:

	组数量	组索引位数 Bit	标签位数 Bit	偏移位数 Bit
1	32	5	21	6
2	8	3	23	6
3	4	0	26	6
4	256	8	18	6
5	64	6	19	7
6	256	8	18	6
7	64	6	20	6
8	32	5	20	7

10. 解:

$$(1) 0.22(1-P_1) + 100P_1 < 0.52(1-P_2) + 100P_2$$

$$99.78P_1 - 99.48P_2 - 0.3 < 0 \text{ 时}$$

A 优于 B

$$(2) 0.22(1-P_1) + 0.22KP_1 < 0.52(1-P_2) + 0.52KP_2$$

$$0.22(K-1)P_1 - 0.52(K-1)P_2 - 0.3 < 0 \text{ 时}$$

$$0.22P_1 - 0.52P_2 < \frac{0.3}{K-1} \text{ 时}$$

A 优于 B

11. 解:

地址	直接映射	直接结果	2way	2way结果	4way	4way结果	8way	8way结果
0x1001	0x-b 100-0001	mis/强制	0x-b-b 100-0-001	mis/强制	0x-b-b 100-00-01	mis/强制	0x-b-b 100-000-1	mis/强制
1005	100-0101	mis/强制	100-0-101	mis/强制	100-01-01	mis/强制	100-010-1	mis/强制
1021	102-0001	mis/冲突	102-0-001	mis/强制	102-00-01	mis/强制	102-000-1	mis/强制
1045	104-0101	mis/冲突	104-0-101	mis/强制	104-01-01	mis/强制	104-010-1	mis/强制
1305	130-0101	mis/冲突	130-0-101	mis/冲突	130-01-01	mis/冲突	130-010-1	mis/强制
2ee5	2ee-0101	mis/冲突	2ee-0-101	mis/冲突	2ee-01-01	mis/冲突	2ee-010-1	mis/强制
1105	110-0101	mis/冲突	110-0-101	mis/冲突	110-01-01	mis/冲突	110-010-1	mis/强制

∴ 直接映射发生5次替换, 2路3次, 4路3次, 8路0次

12. 解: 数组地址从 0x0000 ~ 0x017c

偏移位4位, A索引位^{16:47}3位, B索引位^{17:47}4位

对B: 第1次循环命中了 $96 \times \frac{3}{4} = 72$ 次, 第2-100次循环命中了 $0x0080 \sim 0x017c$ 全命中, 命中率 $\frac{72+99 \times 80}{96 \times 100}$

对A: 所有循环都命中了 $96 \times \frac{3}{4} = 72$ 次, 命中率 $\frac{72 \times 100}{96 \times 100} = 75\%$

A 75%, A 缺失率 25%, B 缺失率 16.75%

13. 解.

```

for(int j=0; j<128; ++j)
    for(int i=0; i<64; ++i) {
        A[i][j] = A[i][j+1];
    }

```

14. 解:

(1) 共有128个块, 一个块中可存8个元素

未优化版本, 地址访问不连续, 全miss, 缺失次数 $128 \times 64 = 8192$ 次

优化版本, 每隔16次填满cache, 其中每16个cacheline命中1次, 所以缺失 $128 \times 64 \times \frac{15}{16} = 7680$ 次

(2) 全相联共有128路, 未

优化前每次J循环可填满cache, i跑8次cache更新, 缺失次数 $128 \times 64 \times \frac{1}{8} = 1024$ 次

优化后, 缺失次数为 $128 \times 64 \times \frac{1}{8} = 1024$ 次

(3) 未优化版本, 相邻两个j相差8个cacheline地址, 要出现强制缺失, 此时块数目应为 $128 \times 8 = 1024$ 块, 大小为 $1024 \times 32B = 32KB$

优化版, J每隔16次填满cache, 所以需要 $128 \times 128 \div 16 = 1024$ 块, 总大小 $1024 \times 32B = 32KB$

15.

input

Output

列0 列1 列2 列3

列0 列1 列2 列3

行0 miss hit hit hit

miss miss miss miss

行1 miss hit hit hit

miss miss miss miss

行2 miss hit hit hit

miss miss miss miss

行3 miss hit hit hit

miss miss miss miss

16. 解 ^{ID} input 地址从 $0x0$ ~ $0x1fc$, input ¹ 的地址从 $0x200$ ~ $0x3fc$

1) 组数 $512 \div 16 \div 2 = 16$

1 每循环 64 次填满 cache, 其中每个 cacheline 命中 3 次

∴ 命中率为 $\frac{128 \times 2 \times 3}{128 \times 2} = 75\%$

(2) 不能 ^{增加缓存大小相当于增加组数} 此时 miss 只会部分从冲突 miss 变为强制 miss, 但命中率不变

3) 可以; 当块大小为 32 字节时, 每个 cacheline 可以命中 7 次, 命中率为 $\frac{7}{8} = 87.5\%$ 提高