

6. 这是因为使用地址的中间位作为组索引可以更好地避免缓存抖动。例如，如果有两个变量，它们的地址相差一个块大小，那么如果采用地址高位作索引，则它们会映射在同一组中。但采用中位，就会映射在不同组的高速缓存块中。这样就可以同时存放这两个变量，避免了缓存抖动。

7. 这样做的好处是可以更好地利用虚拟内存系统的页表来管理缓存，减少缓存系统自己维护页表的开销。

$$8. (1) AMAT = 1 \times 97\% + 110 \times 3\% = 4.27 \text{ (周期)}$$

$$(2) 命中率 R = \frac{64KB}{1GB} = \frac{2^6}{2^{20}} = \frac{1}{2^{14}}$$

$$AMAT = 1 \times \frac{1}{2^{14}} + 110 \times \left(1 - \frac{1}{2^{14}}\right) \approx 110$$

(3) 具有局部性而非完全随机访问时，缓存命中率更高，缓存性能更好，而完全随机时命中率很低。

$$(4) R + 110(RR) < 105$$

$$R > \frac{105}{110} \approx 0.955$$

9. 组数 组索引位数 / Bit 标签 / Bit 偏移位 / Bit

1	32	5	21	6
2	8	3	23	6
3	1	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. (1)  $A = 8KB$  直接映射 L1 cache  $\bar{T}_1 = 0.22 + p_1 - 100$   
 $B = 64KB$  固定组相联 L1 cache  $\bar{T}_2 = 0.52 + p_2 - 100$

若  $T_A < T_B$  则  $p_1 - p_2 < 0.003$ ,  
(2)  $T_A' = 0.22 + 0.22kp_1$  (ns)  
 $T_B' = 0.52 + 0.52kp_2$  (ns)

则  $p_1 < \frac{26}{11}p_2 + \frac{15}{11}$

11. ① 直接 (1 路):  $0x001 \rightarrow 1$   
 $0x005 \rightarrow 5$   
 $0x101 \rightarrow 1$   
 $0x105 \rightarrow 5$ ,  $0x1305 \rightarrow 5$ ,  $0x2ee5 \rightarrow 5$ ,  $0x1105 \rightarrow 5$  共替换 5 次
- ② 2 路:  $0x1001 \rightarrow 1'$ ,  $0x1005 \rightarrow 5'$ ,  $0x1021 \rightarrow 1''$ ,  $0x1045 \rightarrow 5^2$ ,  $0x1305 \rightarrow 5/5^2$ ,  $0x2ee5 \rightarrow 5^2/5^{11}$ ,  $0x1105 \rightarrow 5^1$ , 共替换 3 次
- ③ 4 路, 共 1 次
- ④ 8 路, 无需发生替换

12. B: 直接  $M_B = \frac{6}{96} = \frac{1}{16}$   
A: 2 路  $M_A = \frac{12}{96} = \frac{1}{8}$

```

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

```

14. (1) 4KB 直接映射 块大小32B 块数  $\frac{4KB}{32B} = 128$

数组  $A[i][j]$   $j = 0 \sim 127$   $i : 0 \sim 63$  先i后j 认为是int型 17元素占4B  
17块放8个元素 则数组特点共  $8 \times 128$  对应块。直接映射情况下：每块  
会被8个内存区域映射到  $\frac{128}{64/8} = 2^4 = 16$  :

故共 128 个位置，优化前：Miss次数  $64 \times 128 = 8192$ ，优化后：Miss次数  
 $8 \times 128 = 1024$

(2) 优化前： $8 \times 128 = 1024$  优化后： $8 \times 128 = 1024$

(3) 优化前： $4KB \times 8 = 32KB$  优化后  $8 \times 128 \times 32B = 32KB$

	input	out put
15	0 1 2 3	0 1 2 3
0	miss miss miss miss	miss hit miss hit
1	hit hit hit hit	miss hit miss hit
2	hit hit hit hit	miss hit miss hit
3	hit hit hit hit	miss hit miss hit

16. (1)  $\frac{128}{16} = 8$   $\frac{16}{4} = 4$   $32 \times 4 = 128$  最多有128项

所以在一次放入在本4次访问后还可使用(每个数据仅使用一次)

$$(4-1) \times 4 = 759$$

- (2) 不可以 每个数据仅使用一次，不存在因冲突导致 miss 率增加，所以无法改善命中率
- (3) 可以 缓存所有数据仅使用一次，主要是强制失效，增大块大小可显著降低强制失效概率，提高命中率。