

6. ①使用中间位值作为索引可以更好地利用缓存空间, 增加数据的局部性.

②有助于减少冲突, 较低的位数变化较小

③提供更大的标签空间, 提高缓存的容量.

7. ①可以更好地与内存虚拟内存空间系统的硬件支持集成

②缓存可以直接将虚拟地址的页偏移作为块偏移.

③简化缓存系统的管理.

④提高访问效率和速度

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

$$(2) \quad 1 \text{ GB} \gg 64 \text{ KB}$$

约为 110 周期.

(3) 对于 (1), 平均缓存缺失率为 3%, 有一定局部性  
但是对于 (2), 缓存几乎次次缺失, 因此平均  
延时较高.

$$(4) \quad x \times 110 + (1-x) \times 1$$

$$x \times 1 + (1-x) \times 110 < 105$$

$$\text{故 } x > 4.587\%$$

9. 编号 地址位数 缓存大小 块大小 相联度 组数量 组索引位数 标签位数 偏移位数

1	32	4	64	2	32	5	21	6
2	32	4	64	8	8	3	23	6
3	32	4	64	全	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



$$10. 1) \quad 0.22 + 100P_1 < 0.52 + 100P_2$$

$$\text{故 } P_1 - P_2 < 0.003 = 0.3\%$$

$$2) \quad 0.22 + k \cdot 0.22 \cdot P_1 < 0.52 + k \cdot 0.52 \cdot P_2$$

$$\text{故 } 11kP_1 < 15 + 26kP_2$$

11. ~~直接~~ 直接映射, 索引位为 4 位.

$$0x1001 \text{ 模 } 0x000F \text{ 余 } 0x0001 \quad \text{--- miss}$$

$$0x1005 \text{ 模 } 0x000F \text{ 余 } 0x0005 \quad \text{--- miss}$$

$$0x1021 \text{ 模 } 0x000F \text{ 余 } 0x0001 \quad \text{--- 1}$$

$$0x1045 \text{ 模 } 0x000F \text{ 余 } 0x0005 \quad \text{--- 2}$$

$$0x1305 \text{ 模 } 0x000F \text{ 余 } 0x0005 \quad \text{--- 3}$$

$$0x2005 \text{ 模 } 0x000F \text{ 余 } 0x0005 \quad \text{--- 4}$$

$$0xff05 \text{ 模 } 0x000F \text{ 余 } 0x0005 \quad \text{--- 5}$$

5 次块替换.

2 路组相联, 索引位为 3 位.

$$0x1001 \text{ 模 } 0x0007 \text{ 余 } 0x0001 \quad \text{--- miss}$$

$$0x1005 \text{ 模 } 0x0007 \text{ 余 } 0x0005 \quad \text{--- miss}$$

$$0x1021 \text{ 模 } 0x0007 \text{ 余 } 0x0001 \quad \text{--- miss}$$

$$0x1045 \text{ 模 } 0x0007 \text{ 余 } 0x0005 \quad \text{--- miss}$$

$$0x1305 \text{ 模 } 0x0007 \text{ 余 } 0x0005 \quad \text{--- 1}$$

$$0x2005 \text{ 模 } 0x0007 \text{ 余 } 0x0005 \quad \text{--- 2}$$

$$0xff05 \text{ 模 } 0x0007 \text{ 余 } 0x0005 \quad \text{--- 3}$$

3 次块替换.



4路组相联, 索引位为2位.

0x1001余 0x0001, 0x1005余 0x0001

0x1021余 0x0001, 0x1045余 0x0001

全都余 0x0001

3次块替换.

8路组相联 则没有块替换.

12. 缓存A为8个组, 每个组有2路.

缓存B为16个组, 每个组有1路.

一共调回存储类  $100 \times 96 = 9600$  次

对于缓存A, 循环调回了100次完整数组.

数组 array 可分为  $2^4$  个块.

对于缓存A.

~~缺失块数~~ 为

$$1 \left\{ \begin{array}{l} 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \\ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \\ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \\ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \\ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \\ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \end{array} \right.$$

$$2 \left\{ \begin{array}{l} 0 \ 1 \ 2 \ 3 \\ 0 \ 1 \ 2 \ 3 \\ 0 \ 1 \ 2 \ 3 \\ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \ 1 \times 3V \end{array} \right.$$

$\therefore$  为  $25\%$

对于缓存B.

$$\begin{array}{cccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 \times 3V & & & & & & & & & & & & & & & 1 \times 3V \end{array}$$

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 \times 3V & & & & & \end{array}$$

$$\begin{array}{cccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 \times 3V & & & & & & & & & & & & & & & 4V \end{array}$$

$$\therefore \text{为 } \frac{24 + 16 \times 99}{9600} = 16.75\%$$



```

13. for (int i=0; i<12864; i++) {
    for (int j=0; j<64128; j++) {
        A[i][j] = A[i][j] + 1;
    }
}

```

14. 1) 1个块可装 8个数.

缓存可装 128个块, 128组

访问共  $64 \times 128$  次.

缺失次数 { 优化前:  $64 \times 128 = 8192$  次.

优化后:  $64 \times 128 \div 8 = 1024$  次.

2) 缺失次数 { 优化前:  $64 \times 128 \div 8 = 1024$  次

优化后:  $64 \times 128 \div 8 = 1024$  次.

3) 优化前需要: 32KB

优化后需要 4KB.

input 数组

output 数组.

15.  $z_0, z_1, z_2, z_3$   $z_0, z_1, z_2, z_3$

行 0 miss miss hit miss miss miss miss miss

行 1 miss hit miss hit miss miss miss miss

行 2 miss miss hit miss miss miss miss miss

行 3 miss hit miss hit miss miss miss miss

一个缓存可放 2个块, 2组, 一组一路.

一个块可以放 4个数据.



16. 缓存有 16 组, 每组 2 路.

数组 input 2 行, 每行 128 个数, 分为 32 个块

111. 块 1.	块 2.	...	块 16	...	块 32
1x3V	1x3V		1x3V		1x3V
块 33.	块 34.	...	块 48.	...	块 64
1x3V	1x3V		1x3V		1x3V

命中率为 75%

12) 增加缓存大小, 增加组数, 不能

命中率依旧为 75%

13) 增加块大小, 增加了每个块容纳的数据量  
可以增加命中率.