

6. 标签的作用是判断当前索引的块是否为目标块(CPU发出的物理地址所对应的块)。引索的功能是定位可能是目标块的缓存块。

组索引的位数决定了缓存的组数目。选择地址的中间位作为组索引主要是为了实现较好的数据分布和并行性。中间位提供了足够的位数来表示一定数量的组，从而避免数据集中，减少冲突提高缓存利用率。

高位作为标签确保每个缓存行都有一个唯一的标识，实现准确的缓存命中与缓存替换。

如果用高位做为组索引，中间位做为标签会出现有限的引索位数，数据冲突增加；标签位数的增加会引发误判；缓存利用率，命中率下降等问题。

7. ①. 简化地址转化，地址的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，可使用相同的位字段进行地址转换，简化硬件设计，节约成本。

②. 提高地址定位效率，无需额外的位操作和运算，加快访问速度。

③. 支持高效的虚拟内存映射，相同的位数更有效地利用虚拟内存空间。

④. 提高缓存的一致性。

$$8.(1). 1 \times 97\% + 1/10 \times 3\% = 4.27$$

$$(2). 命中率为 \frac{64KB}{1GB} = \frac{2^6 \times 2^{10}}{2^{30}} = 2^{-14}$$

$$1 \times 2^{-14} + 1/10 \times (1 - 2^{-14}) \approx 1/10$$

(3). 程序局部性越强，缓存发挥的作用越好，访存性能越好

$$(4). 1 \times P + 1/10(1-P) = 1/10 - 1/10P < 1/10 \quad P > 4.6\%$$

命中率P需大于4.6%



9.

编号	地址位数 Bit	块大小 KB	块数 Byte	相联度	组数量	组内索引位数 Bit	标签位数 Bit	偏移位数 Bit
1	32	4	64	2	32	5	1	6
2	32	4	64	8	8	3	3	6
3	32	4	64	全相联	1	0	6	6
4	32	16	64	1	256	8	0	6
5	32	16	128	2	64	6	1	7
6	32	64	64	4	256	8	2	6
7	32	64	64	16	64	6	4	6
8	32	64	128	16	32	5	4	7

10. (1)  $0.22(1-P_1) + 100 \cdot 22P_1 < 0.52(1-P_2) + 100 \cdot 52P_2$

$$\therefore P_1 - P_2 < 0.003$$

(2)  $0.22(1-P_1) + k \cdot 0.22P_1 < 0.52(1-P_2) + k \cdot 0.52P_2$

$$0.22P_1 - 0.52P_2 < \frac{0.3}{k-1}$$

11. 块数 mod 16 分别为 1, 5, 1, 5, 5, 5, 5

∴ 直接映射: 5 次

2 路组相联: 3 次

4 路组相联: 3 次

8 路组相联: 0 次



扫描全能王 创建

12. 16 byte 可包含 4 个 int 型变量，每个缓存有 16 个块

array[96]，需要 24 个块，采用 LRU 替换策略

缺失率：缓存 A： $1 \times \frac{1}{4} = \frac{1}{4}$  2 路组相联，使 0~7, 8~15, 16~23，同时竞争

缓存 B： $\frac{2}{3} \times \frac{1}{4} = \frac{1}{6}$  直接映射，0~7, 16~23 时竞争

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

14. (1) 有 128 块，每块有 8 个 int 型变量

优化前，每次都读缺失，共  $2^{13}$  次

优化后，每 8 次缺失一次，共  $2^{10}$  次

(2). FIFO 替换，则优化前后都为  $2^{10}$  次

(3) 优化前  $2^{13}/2^3 = 2^{10}$  块 32KB 缓存

优化后 1 块 32B 缓存

```
int input[4][4], output[4][4];  
for (int i=0; i<4; ++i){  
    for (int j=0; j<4; ++j){  
        output[j][i] = input[i][j];  
    }  
}
```

每块含 4 个 int 变量

15

input 数组

	列 0	列 1	列 2	列 3
行 0	miss	✓	✓	✓
行 1	miss	✓	✓	✓
行 2	miss	✓	✓	✓
行 3	miss	✓	✓	✓

output 数组

	列 0	列 1	列 2	列 3
miss	miss	miss	miss	miss
miss	miss	miss	miss	miss
miss	miss	miss	miss	miss
miss	miss	miss	miss	miss



扫描全能王 创建

16. (1).  $\frac{2^9}{2^4} = 32$  块，每块  $\frac{2^4}{2^2} = 4$  个 int 变量  
4 次一次 miss 命中率为 75%

(2) 不能 miss 为强制缺失, 不能通过增加内存总大小改善

(3) 可以，增大缓存的块太小能减少强制缺失，从而提高命中率

