6.

①高位组索引、中间位标签的方案会导致组的数量很少，由于高位的位数少，使得组内缓存行数多，冲突更多

②中位索引、高位标签可以实现较好的均衡。中间位作索引将地址分为多个组，每组中缓存行数少，减少冲突；

高位作标签可以实现较好的地址映射，提高命中率。

7.

①方便地址转换，不需要额外的位移运算或保存多余的位信号。

②提高命中率、CPU访问时提高命中之前访问过的数据的几率

③减少物理内存占用，避免虚拟到物理的映射浪费额外内存空间

8.

1) $1+3\% \times 110 = 4.3$

2) $= \text{Hit time} + \text{Mxp} = 1 + \left(1 - \frac{64kb}{1Gb}\right) \times 110 = 114.96$

3) 1)中程序的缺失率为3%，说明程序有一定的局部性

2)中程序访问的数据完全随机，没有局部性可言，使用缓存后，系统的访问延时也提高

4) 程序的平均缓存命中率需要满足如下条件，

$105 > 1 + (1-p)\,110$

$1-p < \frac{104}{110}$

$p > \frac{6}{110} = \frac{3}{55}$

共21位.

9.

| 编号 | 组数量 | 组索引位数 | 标签位数 | 偏移位数 |
|---|---|---|---|---|
| 1 | 64*32 | 5 | 21 | 6 |
| 2 | 64*8 | 3 | 23 | 6 |
| 3 | 64*1 | 0 | 26 | 6 |
| 4 | 256 | 8 | 18 | 6 |
| 5 | 128*64 | 6 | 19 | 7 |
| 6 | 1024*256 | 8 | 18 | 6 |
| 7 | 1024*64 | 6 | 20 | 6 |
| 8 | 512*32 | 5 | 20 | 7 |

10. A 8kb直接映射缓存 命中延时 0.22ns 缺失率 $P_1$

64kb四路 0.52ns $P_2$.

1)

$$0.22 + 100P_1 < 0.52 + 100P_2$$

$$P_1 < 0.3 + P_2$$

2)

$$0.22 + K_1 \cdot 0.22 \cdot P_1 < 0.52 + K \cdot 0.52 \cdot P_2$$

$$P_1 < \frac{30}{22K} + \frac{52P_2}{22}$$

11. 将需要访问的地址转化为十进制后得 4097、4401、4137、4165、4869、12005、65285.

①直接映射.

缓存有16个块. 每块对应的地址范围 64b.

需要4位作为索引位. 块地址中为 $K \times 1024 + m$ 的块都共享第 m 位. m取0~15.

将上面各数对16取余的结果为 1, 5, 9, 5, 5, 13, 5.

替换3次.

② 二路组相联.

缓存中共有8个组，每个组分2路.

将各地址对8取余后得 1, 5, 1, 5, 5, 5

2次替换

③ 四

4 4

对4取余得 1, 1, 1, 1, 1, 1, 1

3次替换.

④ 八

2 8

0次替换

12. 块大小16b，总容量256b 共有16个位置.

13. 缓存A 2路组相连 共有8组 每组分2路

缓存B 直接映射 16 1

LRU替换（近期最少使用）

代码对96行100列的数组进行赋值 共访问了9600个内存地址. 每个元素4字节，共38400字节内

A: 缓存满时，每个组中的两个缓存行都可能被替换.

缺失率为100%

B: 直接映射，缓存共有256个字节，因此共有64个缓存行，每行可以存储1个数组中的元素. 共访问9600个元素，因此每个元素访问次数为150.

13. 按照列优先的方法优化得

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

14. 整形int型数组一个元素占4字节

1) 4Kb的缓存 块大小为32b 共有128块 ①优化前 由于数组中的元素在内存中实际上是顺序排列的. 每次循环、操作数都不在缓存中,缺失次数为8192次
一块中可以存储8个元素. ~~共有64×128=8192个元素,一共访问了8192×4=32768字节的内存~~
~~32768/32=1024次缓存缺失.~~ ②优化后. 每四次发生一次缺失.
共为2048次

2) FIFO 先进先出 全相连缓存

①优化前 首轮对j从0~128的循环,全都缺失,计128次;第2~4轮对j的循环不缺失.
第5…缺失;第6~8轮不缺失… 共计 $128\times\frac{4}{4}=2048$次.

②优化后. 2048次.

~~3) 至少需要8×32=256个字节.~~

优化前:块大小为32b,则每块可以存储数组中的4个元素.

优化前:需要至少128块, 容量 128×32=4kb

优化后:…?

缓存共32b 一块16b 含4个元素

15. 矩阵转置.

int 4b input 起始 0x0 行上连续.

outpict 0x40

| | input | | | | outpict | | | |
|---|---|---|---|---|---|---|---|---|
| | 列0 | 列1 | 列2 | 列3 | 列0 | 列1 | 列2 | 列3 |
| 行0 | miss | ✗ | ✗ | ✗ | miss | ✓ | ✓ | ✓ |
| 行1 | x | x | x | x | x | ✓ | ✓ | ✓ |
| 行2 | x | x | x | x | x | ✓ | ✓ | ✓ |
| 行3 | x | x | x | x | x | ✓ | ✓ | ✓ |

16.

1) 缓存为2b. 块大小16b 共32个缓存块. 每块可以存储4个元素

i=0时命中率为不命中

i=1,2,3时, 命中

i=4时 不命中

5,6,7 命中

..

命中率 ~

2) 提高: 缓存的总大小越大, 可以缓存的元素越多, 命中率高

3) 不提高: 缓存块大小增加, 每个块可以缓存的元素减少, 命中率降低.