6. 增加组索引可储存的数量，提高缓存效率
标签长度适当，适合缓存容量

7. 方便进行地址转换，优化地址的对齐性能
提高缓存命中率

8. 1) $97\% \times 1 + 3\% \times 110 = 4.27$
2) 命中率 $\frac{64KB}{1GB} = \frac{64}{1\times1024\times1024} \to 0$
110 周期
3) 具有独特访问时间局部性的程序，使用缓存优化策略可以提高
4) $(1-x) + 110x < 105 \quad x < 95.4\%$

10. 1) $0.22p1 + 100((-p1) < 0.52p2 + 100(1-p2)$

$99.78\ p1 > 99.48p2$

$p1 > 0.997\ p2$

2) $0.22p1 + k0.22(1-p1) < 0.52p2 + k0.52(1-p2)$

$0.22(1-k)p1 < 0.3k + 0.52(1-k)p2$

$p1 > \frac{0.3k}{0.22(1-k)} + 2.36p2$

9. 根据给出的不同缓存配置，补全下表中缺失的字段。

编号	地址位数 Bit	缓存大小 KB	块大小 Byte	相联度	组数量	组索引位数 Bit	标签位数 Bit	偏移位数 Bit
1	32	4	64	2	32	5	21	6
2	32	4	64	8	8	3	23	6
3	32	4	64	全相联	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

11. (addr) mod (16 × 相联)

0x1001	1	1	1	1
1005	5	5	5	5
1021	1	1	21	21
1045	5	5	5	45
1305	5	5	5	5
2ee5	5	5	25	65
4f05	5	5	5	5
	1	2	4	8
	5	3	1	0
	16	16×2	16×4	16×8
	0x10	0x20	0x40	0x80

12. A. $16 \times 8 \times 2 = 256$ tk 组

B. $16 \times 16 = 256$ tk

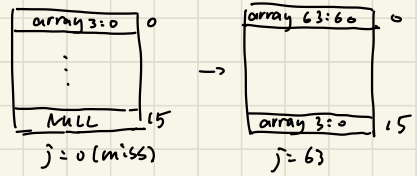
一块 $16 = 4 \times$ int32_t

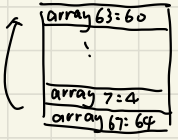
array[96]

$24 \times 4 \times$ int32_t

24 tk

B:

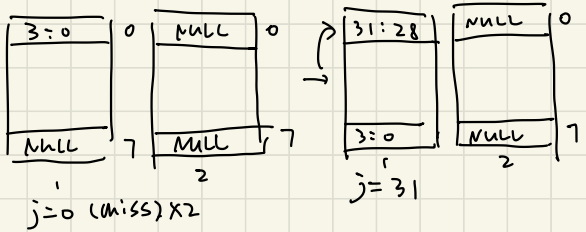


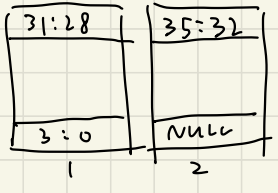


0 miss
1 hit
2 hit
3 hit

miss rate $= \frac{1}{4} = 25\%$

A:





0 miss×2
1 hit
2 hit
3 hit

miss rate $= \frac{2}{5} = 40\%$

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

14. 1) 4KB = 1024 * sizeof(int).

32B = 8 * sizeof(int).

128 tk.

访问 128×64 次

优化前. 内层循环结束 hit 共 64 次

$(128-1) \times 64 = 8128$ 次.

后. $\frac{1}{8} \times 128 \times 64 = 1024$ 次

2) 128行 8× int /

优化前 $j=0$ miss 128

$1 \le i < 8$ miss 0

$8 \le i < 64$ miss $56 \times 128 \times \frac{1}{8}$

> 1024 次

后. $\frac{1}{8} \times 128 \times 64 = 1024$ 次.

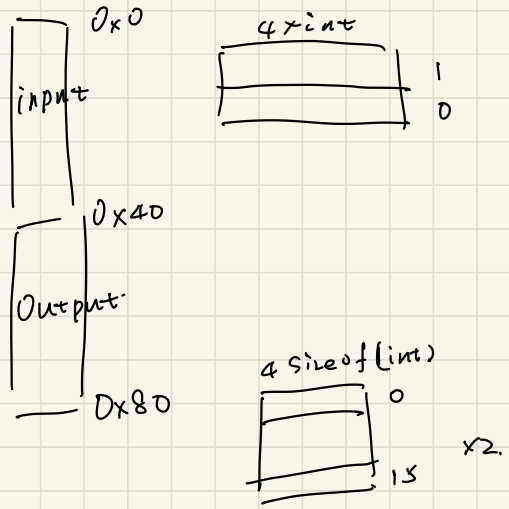
3) $128 \times 64 \times 32 = 2^{18}$ B

15. 考虑如下所示的实现矩阵转置功能的 C 语言代码片段：

```
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];
}
```

假设系统中 int 变量为 4 字节，input 数组的起始地址为 0x0，output 数组的起始地址为 0x40.假设上述程序运行在一个存在 L1 缓存的系统中，该缓存总大小 32 字节，块大小 16 字节，直接映射，缓存策略为直写、写分配。若初始缓存为空，系统仅在对 input 和 output 的读写时会访问缓存。请在下表中填写执行上述程序时数组中每个元素的缓存命中情况。

	input 数组				output 数组			
	列 0	列 1	列 2	列 3	列 0	列 1	列 2	列 3
行 0	miss	hit	hit	hit	miss	hit	hit	hit
行 1	miss	hit	hit	hit	miss	hit	hit	hit
行 2	miss	hit	hit	hit	miss	hit	hit	hit
行 3	miss	hit	hit	hit	miss	hit	hit	hit



512B = 128 x sizeof(int)

16. 1) input

0x0 input[0]　　0x1FF

0x200 input[1]　　0x3FF

但12. miss rate = $\frac{2}{5}$ = 40%

2) 可以. 缓存的大小越大，可以缓存的数据块数量越多，可以缓存的元素越多，减少不命中次数，提高命中率.

3) 适当增大块大小可以提高命中率，但块过大可能导致空间浪费，使得空间利用率下降.