

6. 解：如果使用高位做索引，一些连续的块就会被映射到相同的高速缓存组中，这样程序可能只能用整个高
速缓存的一个组，高速缓存的使用率比较低，而如果以中间位作为索引，相邻的块会被映射到不同的
高速缓存组中，这样对于高速缓存的使用率比较好，更好的发挥其作用。

11. 答：两者相匹配可以消除从虚拟地址到物理地址的复杂转换，缓存查找可以在虚拟空间直接进行，而无
需访问页表和地址转换，提高访存效率。

8. 解：(1) 平均访问延时 $(1-3\%) \times 1 + 3\% \times 110 = 4.27$ 个周期

(2) 平均缓存命中率为 $\frac{64KB}{1GB} = \frac{1}{2^4}$ ，故平均访问延时 $1 + (1 - \frac{1}{2^4}) \times 110 \approx 110$ 个周期

(3) 只有当访问的数据具有局部性时，每次访问数据才能大概率在缓存中找到，即平均缓存缺失率很
低，只有少量数据需要访问主存，提高访存效率。

(4) 平均缓存命中率为 k ， $k \times 1 + (1-k) \times 110 < 105 \Rightarrow k > \frac{5}{109}$

位址位數 Bit	後字大小 KB	块大小 Byte	相聯度	組數量	組索引 bit	標籤 bit	塊內偏移 bit
1 32	4	64	2	32	5	21	6
2 32	4	64	8	8	3	23	6
3 32	4	64	全相聯	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

$$1678(1) \quad 0.22 + p_1 \times 100 < 0.52 + p_2 \times 100 \Rightarrow p_1 - p_2 < 0.003$$

$$(2) \quad 0.22 + p_1 \times 0.22(k-1) < 0.52 + p_2 \times 0.52(k-1) \Rightarrow p_1 < \frac{0.3}{0.22(k-1)} + \frac{26}{11}p_2$$

解：直接映射索引位有4位，16组，则根据块地址，0x100写入0001，0x1005写入0101，0x1021替换0001

`0x1645`替換`0x01`, `0x1305`替換`0x01`, `0x2ee5`替換`0x01`, `0x1f05`替換`0x01`. 故替換 5 次

同理 2路相互联结有3往，8组2路 故替换3次

4路相串联 索引有准 4组 4路 故替换 3次

8路组相联 索引直接 2组 8路 故替换 0次

12. 题：直接映射时，假设每次载入一块即 16Byte, 4 in 32-bit, 因此每访问 4 次数据会缺失一次。

而素有4位，因此復微波失率为 $\frac{1}{4}$

与直接映射同理，每访问4次数据会缺失一次，故缓存缺失率为 $\frac{1}{4}$

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

一共有 128 行

14题 1) 优化前 每次读入缓存 8 个 int, 故读缺失向写命中, 缓存缺失率为 $\frac{1}{2}$, 缓存缺失次数为 $64 \times 128 \times 2 = 8192$ 次
 优化后 每读写 8 次就会有一次读缺失, 故缓存缺失率为 $\frac{1}{8}$, 缓存缺失次数为 $\frac{64 \times 128 \times 2}{8} = 1024$ 次

2) 若全相联, 每次载入 8 个 int, 当遍历一次后, 则会将缓存 128 个块, 因此当计时正好命中 5 次
 优化前 读写 8 次会缺失 1 次, 缓存缺失次数为 $\frac{64 \times 128 \times 2}{16} = 1024$ 次

优化后 缓存工作情况与直接映射时相同, 故缓存缺失次数为 1024 次

3) 优化前所需缓存容量为 $128 \times 8 \times 32 = 2^15 B = 32KB$

优化后 所需缓存容量为 $128 \times 64 \times 4 = 2^15 B = 32KB$

15题:

input				output			
列 0	列 1	列 2	列 3	列 0	列 1	列 2	列 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

