

第十三周作业

6. 相差一个块大小的两个变量，如采用高位作索引，会映射在同一组中，而采用低位，则会映射在不同组的高速缓存块中。因此采用中间位可有效避免冲突，利用好多路的特性。

7. 这样可以使缓存的一组对应于虚拟内存的一页，在进行读写操作时，缓存的数据大小与虚拟内存一页相对应，组索引和内存映射可以同步进行，减少命中时间。即缓存不等待TLB完成地址翻译，直接利用地址翻译前后不变的索引进行索引，再将标签与翻译后物理地址对应位比较，实现速度提升。

8. 1) $1 \times 0.97 + 110 \times 0.03 = 0.97 + 3.3 = 4.27$

$\therefore 4.27$ 个周期

2) $64KB / 1GB \approx 0.00006$

$1 \times 0.00006 + 110 \times 0.99994 = 109.99346$

$\therefore 109.99346$ 个周期

3) 局部性原理使得缓存的命中率提升，其局部性相关越差，其 miss 率越高，性能提升越低。

4) 设命中率为 X

$$1 \cdot X + 110(1-X) \leq 105$$

$$X + 110 - 110X \leq 105$$

$$109X \geq 5$$

$$X \geq 0.04587156$$

大于 4.587156%

红书 题型 分类

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
		Bit	KB	Byte	Bit	Bit	Bit
1	32	4	64	2	32	31	6
2	32	4	64	8	18	23	6
3	32	4	64	全相联	1	26	6
4	32	16	64	1	256	11	6
5	32	16	48	2	14	0	7
6	32	64	64	4	256	8	6
7	32	64	64	16	64	6	20
8	32	64	128	16	32	5	7

10. (1) 系统A: 8KB直接映射缓存 命中延时 0.22ns 缓存缺失率 P_1 缺失代价 100ns

$$0.22 + P_1 \cdot 100$$

系统B: 64KB四路组相联缓存 命中延时 0.52ns 缓存缺失率 P_2 缺失代价 100ns

$$0.52 + P_2 \cdot 100$$

$$A \text{ 优于 } B \quad 0.22 + 100P_1 < 0.52 + 100P_2$$

$$100P_1 < 0.3 + 100P_2$$

$$P_1 - P_2 < 0.003$$

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

$$0.22kP_1 - 0.52kP_2 < 0.3$$

		直接	2路	4路	8路
11.	0x100 / 0001 0000 0000 0001	0001 ✓	001 ✓	01 ✓	1 ✓
	0x1005 / 0001 0000 0000 0101	0101 ✓	101 ✓	01 ✓	1 ✓
	0x1021 / 0001 0000 0010 0001	0001 ①	001 ✓	01 ✓	1 ✓
	0x1045 / 0001 0000 0100 0101	0101 ②	101 ✓	01 ✓	1 ✓
	0x1305 / 0001 0011 0000 0101	0101 ③	101 ①	01 ①	1 ✓
	0x2ee5 / 0010 1110 1110 0101	0101 ④	101 ②	01 ②	1 ✓
	0xff05 / 1111 1111 0000 0101	0101 ⑤	101 ③	01 ③	1 ✓
		5次	3次	3次	0次

12. A 2路组相联 块大小16字节 总容量256字节 LRU替换
 B 直接映射 块大小16字节 总容量256字节 LRU替换

32bit 低5位为块内偏移 $256 \div 16 = 16$

直接映射 中间4位为索引

2路组相联 中间3位为索引

一个缓存块可存入4个数据

① 直接映射

j[0] - j[3] 存入 N=0 miss 1次 成功 3次

j[0] - j[3] 存入 N=1 miss 1次 成功 3次

j[4] - j[7] 存入 N=2 miss 1次 成功 3次

j[56] - j[59] 存入 N=15 miss 1次 成功 3次

j[60] - j[64] 存入 N=0 miss 1次 成功 3次

地址是关键

①直接映射：96 分为 $1^0 32$ $2^0 32$ $3^0 32$ ★一定注意与余数有关

共有 64 组 $32-63$ 不会被替换 $32 \bmod 64 = 32$

$$\frac{1}{4} \times \frac{2}{3} = \frac{1}{6}$$
 96 \bmod 64 = 32

对应于相同部分

② 2 路组相联 不停交替

$$\frac{1}{4}$$

13. `for (int j=0; j<128; ++j) {`

`for (int i=0; i<64; ++i) {`

$$A[j][i] = A[j][i] + 1;$$

{ 逐一用取 } 读取的顺序不应该是从内到外，而是从外到内

14 (1) $4KB / 32B = 128$ 块放 $32 \div 4 = 8$ 个数据

① 优化前 内层为 128 每次均 miss

$$128 \times 64 = 8192 \text{ 次}$$

② 优化后

内层每 8 次 miss 1 次，且内层为 128，故每外层循环一次相同

$$8192 \div 8 = \underline{\underline{1024}} \text{ 次}$$

(2) 改用 FIFO 策略

优化前: $128 \times 64 \times \frac{1}{8} = 1024$

优化后: $128 \times 64 \div 8 = 1024$

(3) 优化前: $128 \times 64 \times 32 = 262144$ 字节 $128 \times 64 \div 8 \times 32 = 32$ KB

优化后: 只要有一组即可 32 字节

15.

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	miss	hit	miss	miss	miss	miss	miss	miss
行1	miss	hit	miss	hit	miss	miss	miss	miss	miss
行2	miss	miss	hit	miss	miss	miss	miss	miss	miss
行3	miss	hit	miss	hit	miss	miss	miss	miss	miss

16.(1) $512 \div 16 = 32$ $16 \div 4 = 4$

$32 \times 4 = 128$ 每完成一次外，最多存 128 项

所以不存在一次放入在本 4 次访问后还可使用 (每个数据仅使用一次)

$14 - 1 \div 4 = 75\%$

(2) 不可以 我们每个数据仅使用一次，存在因冲突导致 miss 率增加，
所以无法改善命中率

(3) 可以 我们所有数据仅使用一次，主要是强制 ~~冲突~~，增大块大小
可显著降低强制冲突失效比率，提高命中率