

6. ① 组索引用于确定数据在缓存中的组或集合位置, 通过使用地址的中间位作为组索引, 可以将缓存数据均匀地分布在不同的组中, 这样有利于避免冲突。如果使用高位作为组索引, 取值范围小, 组数少, 容易冲突, 降低缓存性能。

② 高位的位数比中间位多, 高位作为标签可以支持更大的缓存容量

7. 好处: ① 简化地址转换, 共享相同的位字段, 减少地址转换过程中的计算

② 提高地址解码效率: 用相同的位字段来直接定位缓存中的 ^{和操作} 组索引和块内偏移, 无需进行额外计算

③ 兼容性和一致性

实现虚拟内存系统和缓存系统之间的

8. 解: ① $(1 \times 97 + 110 \times 3) \div 100 = 4.27$ 周期

② $\because 1GB > 64KB$, $L1$ 缓存无法容纳整个数组, 每次访问都会导致缓存缺失, 访问延时为 110 周期

③: ① 中, 访问命中率 97%, 说明程序在一段时间内会多次访问已经访问过的数据, 即时间局部性, $L1$ 缓存的存在可以加快对命中数据的访问速度, 从而提高缓存性能。② 中, 由于数据访问是完全随机的, 局部性原理无法利用到, $L1$ 缓存无法发挥作用, 直接访问主存, 导致访问性能下降。

④ 设缓存命中率为 $x\%$, $(1 \times x + 110 \times (100 - x)) \div 100$
 $\therefore x > 4.587$

\therefore 只要缓存命中率大于 4.587%, $L1$ 就能获得性能收益

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

10. 解 (1) A 时间: $0.22(1-p_1) + p_1 \times 100 = 0.22 + 99.78p_1$

B 时间: $0.52(1-p_2) + p_2 \times 100 = 0.52 + 99.48p_2$

$$0.22 + 99.78p_1 < 0.52 + 99.48p_2$$

$$\Rightarrow p_1 < 0.003 + 0.997p_2$$

(2) A 时间: $0.22(1-p_1) + 0.22kp_1 = 0.22 + 0.22(k-1)p_1$

B 时间: $0.52 + 0.52(k-1)p_2$

$$0.22 + 0.22(k-1)p_1 < 0.52 + 0.52(k-1)p_2$$

$$\Rightarrow p_1 < \frac{1.36}{k-1} + 2.36p_2$$

11. 直接映射: 0次

4路组相联: 4次

2路组相联: 2次

8路组相联: 7次

12. 缓存 A: 缺失率 = $100 \times 7 / 9600 = 7.29\%$

缓存 B: 缺失率 = $700 / 9600 = 7.29\%$

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

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

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

}

14. (1) 优化前: $128 \times 64 = 8192$ 次

优化后: 128 次

(2) 优化前: 0 次

优化后: 128 次

(3) 优化前: $64 \times 4 / 32 = 8$ 个块

优化后: $256 / 32 = 8$ 个块

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

16. (1) $512 \div 16 = 32$ 块, 1 块可以存储 2 个元素, 且两个元素属于不同的组

\therefore 命中率为 100%

(2) 不会。因为程序访问率 input 是按行访问, 每次访问都会读取 2 个元素, 而缓存的块大小已经足够存储两个连续的元素, 增加缓存的总大小并不会改变程序的访问模式和访问方式, 因此对命中率没有影响。

(3) 可以。增加块大小, 使得每个块存储更多元素, 从而提高命中率。

17. (1) \therefore 虚拟地址 0x05a4

\therefore 页号 0x05, 偏移量 0xa4

\therefore 组号为 1, 标签为 0x05 命中, 物理地址 0x0D ^{组号} 0xa4 ^{得 0x0DA4}

(2) $2^{14-6} = 2^8 = 256$ 个条目

(3) 物理地址 0x0DA4 对应页号 0x0D, 没有块偏移, 有效位为 0, 命中缓存

18. (1) A B C D A B C D

way 0 - A A A A A A A 命中率 50%

way 1 - - B B B B B B

命中? N N Y Y Y Y Y Y

(2) 使用 LRU, 命中率 75%

19. (1) 原因: 同一缓存组内的不同块需要通过低位标签来进行区分和匹配

(2) 影响: 影响替换策略, 影响缓存性能

(3) $8kB / 4 = 2kB$, $\log_2(2kB) = \log_2(2048) = 11$ 比特

20. 监听一致性优点: 实现简单, 通信开销较低

缺点: 总线瓶颈, 无法利用局部性原则

目录一致性优点: 通信开销较低, 利用了局部性原则

缺点: 实现复杂, 延迟增加

缓存一致性的实现代价体现在: 软硬件开销, 通信开销, 一致性维护开销