

作业12

C4

6. 组索引采用地址的中间位, 可以充分利用现代计算机中多级缓存的特点, 各级缓存的组数比较小, 使得组索引不占用过多位数, 同时也保证各缓存组的数据量相对均衡, 避免了某些缓存组的负载过重或过轻, 影响缓存性能;

标签采用地址的高位, 可以保证标签的位数相对较多, 使得缓存中的数据块可以被更细粒度地分区管理, 提高了命中率和缓存利用率。

现行方案是合理的缓存映射策略, 可以保证缓存性能的同时充分与利用各级缓存的资源。

7.  $4096 + 4096 + 4096 + 4096 + 4096 + 4096 + 4096 + 4096$

① 实现统一的地址映射, 减少映射次数

② 页表机制可以减少物理内存的开销, 并使得多个进程之间可以共享虚拟地址, 而位数相同可以充分利用页表机制, 在保证地址映射正确性的情况下提高程序访存效率

③ 简化内存管理的实现, 减轻操作系统的负担

8.

$$1) 110 \times 3\% + 1 \times 97\% = 4.27$$

平均延时 4.27 周期

$$2) 64 \text{ KB} / 1 \text{ GB} = 2^{-14}$$

$$110 \times (1 - \frac{1}{2^{14}}) + 1 \times \frac{1}{2^{14}} = 109.99$$

平均延时 109.99 周期

3) 缓存的局部性越强

即存储器的内容越少,

平均缓存缺失率越高,

访存性能越差

4) 命中率  $\eta$

$$110 \times (1 - \eta) + 1 \times \eta = 105$$

$$\eta = 4.587\%$$

平均缓存命中率至少为 4.587%





9.	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
编号	Bit	KB	Byte			Bit	Bit	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	20	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	21	7

10.  $0.22 \times 10^3 + (1 - 0) \times 0.11$

$$1) 0.22 \text{ ns} \times 1 + 100 \text{ ns} \times p_1 = 1$$

$$< 0.52 \text{ ns} \times 1 + 100 \text{ ns} \times p_2$$

$$\text{解出 } p_1 - p_2 < 0.3\%$$

$$\text{条件: } p_1 - p_2 < 0.3\%$$

$$2) 0.22 + 0.22k p_1 < 0.52 + 0.52k p_2$$

$$0.22 p_1 - 0.52 p_2 < \frac{0.3}{k}$$

$$\text{条件: } 0.22 p_1 - 0.52 p_2 < \frac{30}{k} \%$$

11.  $0.22 \times 10^3 + (1 - 0) \times 0.11$

块地址标记为相对  $0 \times 100$  的相对地址

$$+ 0 + 4 + 20 + 44 + 304 + 1004 + 6404$$

直接映射: 16个组 5次块替换

2路组相联: 8个组 5次块替换

4路组相联: 4个组 3次块替换

8路组相联: 2个组 0次块替换





12.

 $256 / 16 = 16$  缓存有16个块

int 32\_t 占4字节

则一个块存4个 int 32\_t

 $96 / 4 = 24$  个 内存中的块

B: 一轮后

 $\begin{matrix} 16 \\ \vdots \\ 23 \\ 8 \\ 15 \end{matrix}$ 
 $\begin{matrix} 0 \\ \vdots \\ 7 \\ 8 \\ 15 \end{matrix}$ 

8个缺失 - 8个缺失

8个命中

后面的缺失率:  $\frac{2}{3}$ 

$$(1 \times \frac{1}{4} + 99 \times \frac{2}{3} \times \frac{1}{4}) / 100 = 16.75\%$$

A: 一轮后

 $\begin{matrix} 08160 \\ \vdots \\ 715237 \\ \times \checkmark \checkmark \end{matrix}$ 

LRU 最近最少使用

缺失率为  $1 \times \frac{1}{4} = 25\%$ 

用A: 2路组相联 缺失率 25%

用B: 直接映射 缺失率 16.75%

13.

for (int i=0; i&lt;128; ++i) {

for (int j=0; j&lt;64; ++j) {

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

}

}

14.

1)  $4\text{KB} / 32\text{B} = 128$  缓存有128个块

int 占4个字节

则一个块存8个 int 值

优化前:  $64 / 8 = 8$  个块  $128 / 8 = 16$ 
 $64 \times 128 = 8192$  次 缺失
优化后:  $64 / 8 \times 128 = 1024$ 

优化前缓存缺失次数: 8192

优化后: 1024

2) 全相联

优化前: 一轮存满128块

 $64 / 8 \times 128 = 1024$ 

优化后: 不变

优化前缓存缺失次数: 1024

优化后: 1024

3) 优化前:  $8 \times 64 / 8 \times 16 + 1 = 129$  块
 $129 \times 32\text{B} = 4,032\text{B} = 4.03125 \text{ KB}$ 

优化后: 8块

 $8 \times 32\text{B} = 256\text{B}$ 

优化前需要的缓存容量: 4.03125 KB

优化后: 256 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

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

16.

1) 缓存有 32 个块, 有 16 个组

1 个块存 4 个 int 数据,  $128/4=32$

两路 满足需求  $= 8 \times 8$

缓存命中率 75%

2) 不可以

本例中只要一个组就足以容纳

3) 可以

块 ↑ 存的 int ↑

缓存读一个块记转的数据 ↑

