

5.16

6. 当缓存处于高速时,若对如  $X[0]$ 、 $X[1]$ 、 $X[0]$ 、 $X[1]$ ... 类型顺序访问时,会发生抖动,当二者不同外同一组时,其索引不同,应存放进入不同索引的地址。若地址高位索引,则索引读取速度~~不~~高速。则无法同时存放在高速缓存下,  $X[0]$ 、 $X[1]$  会存放进同一组。若采用中位,则可映射在不同组的高速缓存中。

7. 因为处理器发出一个读写请求后, 其虚拟地址解析为页虚拟页号和页内偏移两部分, 虚拟页号访问内存中的物理页号, 页内偏移先访问缓存。页内偏移位数与组索引和块内偏移总位数相等时, 更方便定位缓存单元, 位数一一对应。

8. (1) 平均延时  $\text{cycle}_1 = 1 \times 97\% + 110 \times 3\% = 4.27$

平均访问延时为 4.27 个周期

(2) 若完全随机数据, 则命中率为  $\gamma_0 = \frac{64\text{KB}}{1\text{GB}} = \frac{1}{2^{14}}$

延时: 若有缓存则  $\text{cycle}_2 = 1 \times \frac{1}{2^{14}} + (1 - \frac{1}{2^{14}}) \times 110 \approx 110$

无缓存:  $\text{cycle}_2 = 105$

(3) 局部性可以显著提高缓存命中率, 从而减少延时。若完全随机, 即不具有局部性(如2), 则缓存的加速作用无法体现。

(4) 设命中率为  $\gamma_0$

则  $\text{cycle} = 1 \times \gamma_0 + 110 \times (1 - \gamma_0) = 110 - 109\gamma_0$

$110 - 109\gamma_0 < 105 \Rightarrow \gamma_0 > 4.59\%$

命中率大于 4.59% 时即有收益



9. 解:

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	16	4	22	6
3	32	4	64	全相联	64	6	20	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. 由题, A为8KB直接映射, B为64KB回路组相联缓存

$\therefore P_1 > P_2$

(1) A平均延时:  $t_A = 100P_1 + 0.22(1-P_1)$

B:  $t_B = 100P_2 + 0.52(1-P_2)$

$t_A < t_B \therefore 100P_1 + 0.22 - 0.22P_1 < 100P_2 + 0.52 - 0.52P_2$

$\Rightarrow 99.78P_1 - 99.48P_2 < 0.3$

近似为:  $P_1 < P_2 + 0.3\%$  即  $P_2 < P_1 < P_2 + 0.3\%$

(2) A:  $t_A = 0.22KP_1 + 0.22(1-P_1)$

B:  $t_B = 0.52KP_2 + 0.52(1-P_2)$

$t_A < t_B \Rightarrow 0.22 + 0.22(K-1)P_1 < 0.52 + 0.52(K-1)P_2$

$\therefore 2(K-1)P_1 - 52(K-1)P_2 < 30$

即  $11P_1 - 26P_2 < \frac{30}{K-1}$  时 A优于 B



11. <sup>①</sup> ~~0x~~ 0x/00 = 10000 00000000 u)      ② 0x1305 = 1001 1000 0101 (2)

③  $0x1005 = 100000000 \overset{0}{\underset{0}{|}}$  (2)      ④  $0x2ae5 = 1011101110101$  (2)

③  $O_{x/021} = 10000\ 0010\ 0001\ \omega$       ④  $O_{x/f05} = 1111\ 1111\ 0000\ 0101\ (2)$

④  $0x1045 = 1000001001010101_2$

⑩ 直接映射  $\rightarrow$  有 ~~地址~~ 不考虑块内偏移  $\therefore$  选择物理地址后  $\log_2 16 = 4$  位为 index

直接映射: 在 ③、④、⑤、⑥、⑦ 时发生替换, 共 5 次

2路组相联: 在⑤、⑥、⑦时替换, 共3次

4路组相联：在①时替换，共1次

8路组相联: 无替换

12. 2块大小为16  $\therefore$  index 有 8 位,  $256B \div 16 = 16B/\text{块}$   $\therefore$  块内偏移 4 位  
12. 块大小为 16B, index 为 4 位, 块大小 16B, index:  $256 \div 16 = 16$  块, 块内偏移 4 位

12 A: 2路组相联

一个 int 有 4B : 每块可以存 4 个 array[i]

每4个 array [j] 中 有一个 miss,  $\therefore$  命中率 75%

由于  $96 \div 4 = 24 > 16$ ，在第 17 次访问新块时替换第一个块，

但由于无论哪个块,命中率均为 75%  $\therefore A_{\text{miss-rate}} = 25\%$

B: 直接映射

由上, 每个块中  $miss\_rate = 25\%$ .  $96 \div 4 = 24 < 16 \times 2$ , 因此不会块替换.

~~因此在下一个循环~~  $B_{miss\_rate} = 25\%$



13. 改为:

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

14. 读写时发生访问缓存

(1) 4KB, 块大小: 32B, 有 5 位 ~~offset~~ <sup>offset</sup>,  $4KB \div 32B = 128$  块, 7 位 ~~index~~ <sup>index</sup>

int 类型大小为 4B,  $32B \div 4B = 8$  个, 即一个块中有 8 个 A[j][i]

优化前:  $64 > 32$ , 所以每次访问都缺失半, 即缺失  $64 \times 128 = 2^{13}$  次

优化后:  $64 \div 32 = 2$ , 每 64 个周期缺失 8 次, 共缺失:  $8 \times 128 = 2^{10}$  次

(2) 全相联,  $\therefore$  有  $128$  个组

优化前: 由于  $64 > 32$ , 在一个组访问后需至另一个组访问下一个 A[j][i]  
所以缺失仍为  $2^{13}$  次

优化后: 全相联不改变, 每 8 个周期缺失一次,  $\therefore 8 \times 128 = 2^{10}$  次

(3) 直接映射: 块大小仍为 32B

优化前: 设当前块首为 A[j][0], 除必要缺失外无任何缺失,

则下一个 A[j+1][0] 应位于同一块, 同理 A[127][0] 在同一块,

$\therefore$  需  $4 \times 64 \times 128 = 2^{15} B$  (块大小), 总容量  $2^{13} \times 32 = 2^{16} B$

优化后: 当前为 A[j][0], 下一块首应为 A[j+1][0]

$\therefore$  需  $4 \times 64 = 256 B$  / 块, 总:  $256 \times 32 = 2^{13} B$



# 读写访存、写分配、直写、直接映射

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

块大小 16B, 总大小 32B  $\therefore$  有 2 个块

$16B \div 4B = 4$ , 每块中存 4 个整型

16. 读写时访问缓存, int 大小为 4B, input 起始 0x0, input[0][0] 位置为 0x0

(1) 总大小 512B, 块大小 16B, 两路组相联  $\therefore$  有 16 组, 每组 2 路

每块可以存 4 个整型

程序中 sum 在寄存器,  $\therefore$  只有依次读 input[0][0], input[1][0], input[2][0], input[3][0]...

$\therefore$  地址依次为 0x0, 0x80, 0x1, 0x81, ...

对应组别 0, 0, 0, 0, 0, 1, 1, ...

$\therefore$  在每 4 个循环中, 会出现 2 个 miss  $\therefore$  有  $\frac{128}{4} \times 2 = 64$  个 miss

命中率  $1 - \frac{1}{4} = 75\%$

(2) 不能。增加总大小后, 每块仍只能存 4 个整型, 其首位一定仍会 miss, 命中率仍为 75%, 每 4 个整型缺失 1 个

(3) 可以, 增加块大小, 使每个块可以存储更多整型, 每块首位 miss 后其余位命中, 若每块可存 n 个整型,  $n > 4$  时即可命中率提高