

1. 习题6

处理器访存有空间局部性，地址中间位可能变化范围较大，而高位变化较小，所以高位更适合作标签（~~程序高~~），而低位作索引用于区组的不同块。

如果高位作索引，在^{相联}组数较小的情况下，可能使缓存冲突发生次数增多。

2. 习题7

不相联的组数之和与缓存每组的容量相等，页式虚拟存储器每页的大小相等，可以降低硬件逻辑复杂度，使缓存结构和页式虚拟存储器结构更相似，降低硬件逻辑的复杂度。

3. 习题8

$$1) 1 \times (1 - 3\%) + 110 \times 3\% = 4.27 \text{ 周期}$$

$$2) \text{命中率} = \frac{64 \times 2^{10}}{2^{30}} = \frac{1}{2^{14}}$$

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

3) 若程序访存没有局部性，则高速缓存不能减少访存周期数，~~反而~~提高了硬件复杂度反而降低性能。

但若利用程序访存局部性，高速缓存可以极大减少访存周期数，提高性能。

$$4) 1 \times x + 110(1 - x) < 105$$

$$x > \frac{5}{109} \approx 4.6\%$$

即命中率大于 4.6% 才能获得性能收益

4. 习题 9

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

5. 习题 10

$$1) T_A = 0.22 + (1 - P_1) + (100 + 0.22)P_1 = (0.22 + 100P_1) \text{ ns}$$

$$T_B = 0.52 + (1 - P_2) + (100 + 0.52)P_2 = (0.52 + 100P_2) \text{ ns}$$

$$\text{令 } T_A < T_B$$

$$\text{得 } P_1 - P_2 < 0.003$$

$$2) T_A = (0.22 + 0.22k P_1) \text{ ns}$$

$$T_B = (0.52 + 0.52k P_2) \text{ ns}$$

$$\text{令 } T_A < T_B$$

$$\text{得 } 11P_1 - 26P_2 < \frac{15}{k}$$

6. 习题 11 块地址 16 位

直接映射: 组数 16, 索引 4 位, 标签 12 位

2 路组相联: 组数 8, 索引 3 位, 标签 13 位

4 路组相联: 组数 4, 索引 2 位, 标签 14 位

8 路组相联: 组数 2, 索引 1 位, 标签 15 位

$0 \times 1001 = 0001\ 0000\ 0000\ 0001$ (二进制) 请求 ①

$0 \times 1005 = 0001\ 0000\ 0000\ 0101$ ②

$0 \times 1021 = 0001\ 0000\ 0010\ 0001$ ③

$0 \times 1045 = 0001\ 0000\ 0100\ 0101$ ④

$0 \times 1305 = 0001\ 0011\ 0000\ 0101$ ⑤

$0 \times 2ee5 = 0010\ 1110\ 1110\ 0101$ ⑥

$0 \times ff05 = 1111\ 1111\ 0000\ 0101$ ⑦

(1) 直接映射

① 存入, ② 替换, ③ 替换, ④ 替换, ⑤ 替换, ⑥ 替换, ⑦ 替换

①~⑦ 地址全不同, 索引重复较多, 替换次数多, 为 5 次

(2) 2 路组相联

① 存入, ② 存入, ③ 存入, ④ 存入, ⑤ ⑥ ⑦ 替换, 替换 3 次

(3) 4 路组相联

① 存入, ② 存入, ③ 存入, ④ 存入, ⑤ ⑥ ⑦ 替换, ⑧ 替换, 替换 3 次

(4) 8 路组相联

不会发生替换

综上, 直接映射、2 路、4 路、8 路组相联 替换次数分别为 5, 3, 3, 0

7. 习题 12

每个块存 4 个 int 32-bit, 组内偏移 4 位

A: 组数 8, 索引 3 位 B: 组数 16 索引 4 位

$\&array[0] \sim \&array[31]$: 0 ~ 0000 0111 1100

$\&array[32] \sim \&array[63]$: 0000 1000 0000 ~ 0000 1111 1100

$\&array[64] \sim \&array[95]$: 0001 0000 0000 ~ 0001 0111 1100

A: $i=0$ 第 1 路^写 第 2 路^写 第 1 路替换 缺失率 $\frac{1}{4}$

0~31

32~63

64~95

$i=1$ 第 2 路替换 第 1 路替换 第 2 路替换 缺失率 $\frac{1}{4}$

0~31

32~63

64~95

i 继续增加, 缺失率均为 $\frac{1}{4}$

所以缺失率 25%

B: $i=0$; j 从 0~31, 缓存

0~31
X

j 从 32~63, 缓存

0~31
32~63

j 从 64~95, 缓存

64~95
32~63

缺失率 $\frac{1}{4}$

$i=1$, j 从 0~31, 缓存

0~31
32~63

, j 从 32~63 命中,

j 从 64~95 缓存

64~95
32~63

缺失率 $\frac{1}{4} \times \frac{2}{3} = \frac{1}{6}$

i 继续增加, 仍为 $\frac{1}{6}$, 所以缺失率 $\frac{\frac{1}{4} + \frac{1}{6} \times 99}{100} = 16.75\%$

综上, A 缓存缺失率 25%, B 缓存缺失率 16.75%

8. 习题 13

$A[j][i]$, $A[j][i+1]$ 相邻, 把 i 置于内层循环中, 优化后代码如下:

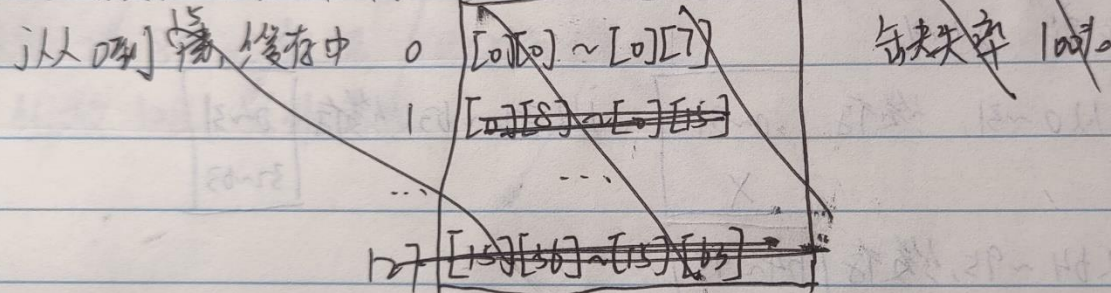
```
for (int j=0; j<128; ++j) {
    for (int i=0; i<64; ++i) {
         $A[j][i] = A[j][i] + 1;$ 
    }
}
```

9. 习题 14

假设 A 定义为 `int A[128][64];`

32 字节块, 每块 8 个 int 数据. 块内偏移 5 位

1) 组数 128, 索引 7 位



从 16 到 31, 优化前: $i=0$

从 0 到 15, 缓存中	0	$[0][0] \sim [0][7]$	缺失率 100%
	1	X	组号 0, 8, 16, ... 写入了数据
	...	X	
	76	X	
	77	$[1][0] \sim [1][7]$	
	

从 16 ~ 127 同理, 缺失率 100%, 覆盖 0, 8, 16, ... 的数据

然后 $i=1$, j 从 0 到 127, 同理, ~~和组号 1, 9, 17, ...~~ 写入数据, 且全部缺失
 i 继续增加, 同理, 所以总缺失次数 $128 \times 64 = 8192$ (读缺失, 写
不缺失)

优化后: $j=0$,

从 0~7, 缺失 1 次

从 8~15, 缺失 1 次, ..., 从 56 到 63, 缺失 1 次

一直到 $j=15$, 均同理

$j=16$ 时, 替换 $j=0$ 存入的数据, 缺失次数与 $j=0$ 同理

j 继续增大, 仍然同理, 所以总缺失次数 $128 \times 8 = 1024$ (读缺失, 写
不缺失)

综上, 优化前缺失 8192 次, 优化后缺失 1024 次

2) ~~优化前~~ 组数 1, 不需要索引, 一共 128 路

优化前: $i=0$

j 从 0 到 ~~127~~¹²⁷, 缓存中 路 0 $[0][0] \sim [0][7]$

路 1 $[10][0] \sim [1][7]$

中...

路 127 $[127][0] \sim [127][7]$, 全部缺失

从 1 到 7, 不缺失

从 8 开始, 与 $i=0$ 同理, 所以总缺失次数 $\frac{64}{8} \times 128 = 1024$

优化后: 每 8 次读取缺失 1 次, 所以总缺失次数 $\frac{1}{8} \times 64 \times 128 = 1024$

综上, 优化前后缺失次数均为 1024 次

3) 优化前: $\frac{8 \times 128 \times 32}{2^{10}}$ KB = 32 KB

优化后: 只需要 32 B

即优化前需要 32 KB 容量, 优化后只需要 32 B

10. 习题 15

每个块 4 个 int 数据, 组数为 2

忽略块内偏移, 二进制表示的地址: 其中最后一位是索引

input[0] 0000 output[0] 0100

input[1] 0001 output[1] 0101

input[2] 0010 output[2] 0110

input[3] 0011 output[3] 0111

	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

11. 习题 16

1) 每个块 4 个 int 数据, 组数为 16, 索引 4 位, 忽略块内偏移, 二进制)

&input[0][0] 000000

&input[1][0] 100000

(忽略块内偏移, 二进制)

&input[0][4] 000001

&input[1][4] 100001

(32 行)

&input[0][24] 011111

&input[1][24] 111111

其中 &input[0][60]: 001111 &input[1][60] 101111

从 0 到 63, input[0][i] 和 [1][i] 存在两路中, 命中率 $\frac{3}{4}$

从 64 到 127, input[0][i] 和 [1][i] 替换原来缓存中内容, 命中率 $\frac{3}{4}$

所以执行上述程序命中率为 75%

2) 增加总大小不能改善命中率

两路组相联, 一个块只能存4个数据, 且并没有数据复用

所以命中率保持 75% 而不会增加

3) ~~可以改善~~ 增加块大小可以改善命中率

保持两路组相联的情况下,增加块大小,每个块存更多 hit 数据,可以增大两次缺失的间隔中的缓存命中次数。