

教材上的作业题，共 6 题： 6.25、6.29、6.34、6.34a（题目如下）、6.36、6.41

（6.25、6.29 直接写答案， 6.34 - 6.41 需写一两行简短的思路。）

（6.34 注：对于写分配的高速缓存，每次写不命中时，需要读取数据到高速缓存中。）

题 6.34a：对于一个总大小为 64 数据字节的高速缓存，重复练习题 6.34。

张梓卫 10235101526

6.25

根据 $C = B * E * S$ ，可以得到 S ，进而分析出 t ， s ， b 的结果

高速缓存	m	C	B	E	S	t	s	b
1.	32	1024	4	4	64	24	6	2
2.	32	1024	4	256	1	30	0	2
3.	32	1024	8	1	128	22	7	3
4.	32	1024	8	128	1	29	0	3
5.	32	1024	32	1	32	22	5	5
6.	32	1024	32	4	8	24	3	5

6.29

这些地址的二进制分别为 1000 0011 0100、1000 0011 0110、1111 1111 1101 分别与块偏移、组索引和标识位对应，据此可以得出结果。

A. 下面的图给出了一个地址的格式(每个小框表示一位)。指出用来确定下列信息的字段(在图中标号出来)：

CO 高速缓存块偏移

CI 高速缓存组索引

CT 高速缓存标记

12	11	10	9	8	7	6	5	4	3	2	1	0
CT	CT	CT	CT	CT	CT	CT	CT	CT	CI	CI	CO	CO

B. 对于下面每个内存访问，当它们是按照列出来的顺序执行时，指出是高速缓存命中还是不命中。如果可以从高速缓存中的信息推断出来，请也给出读出的值。

操作	地址	命中?	读出的值(或者未知)
读	0x834	否	无
写	0x836	是	未知
读	0xFFD	是	0xC0

6.34

Src 是按行访问，dst 是按列访问，根据访问顺序即可得到答案。
若不访问 dst，src 的命中应均为 m h h h，但访问 dst 时，若 $j \% 2 = i \% 2$ ，此时 dst[j] 会替换 src[i] 的缓存行，导致访问不命中。

- 数组 src 从地址 0 开始，而数组 dst 从地址 64 开始(十进制)。
- 只有一个 L1 数据高速缓存，它是直接映射、直写、写分配的，块大小为 16 字节。
- 这个高速缓存总共有 32 个数据字节，初始为空。
- 对 src 和 dst 数组的访问分别是读和写不命中的唯一来源。

对于每个 row 和 col，指明对 src[row][col] 和 dst[row][col] 的访问是命中(h)还是不命中(m)。

例如，读 src[0][0] 会不命中，而写 dst[0][0] 也会不命中。

		dst数组						src数组			
		列0	列1	列2	列3			列0	列1	列2	列3
行0		m	m	m	m	行0		m	m	h	m
行1		m	m	m	m	行1		m	h	m	h
行2		m	m	m	m	行2		m	m	h	m
行3		m	m	m	m	行3		m	h	m	h

6.34a

高速缓存大小 64 字节，可以存储 16 个 int 类型的数，此时有四行高速缓存。

		dst数组						src数组			
		列0	列1	列2	列3			列0	列1	列2	列3
行0		m	m	m	m	行0		m	m	h	h
行1		m	m	m	m	行1		m	h	m	h
行2		m	m	m	m	行2		m	h	h	m
行3		m	m	m	m	行3		m	h	h	h

6.36

数组共： $2 \times 128 \times 4 = 1024$ 字节

A. 假如高速缓存只有 512 字节，显然，每 16 个字节的缓存行能放下 4 个 int。当 x[0][i] 和 x[1][i] 逐次被访问时，指向的缓存行是相同的，故不命中率为 100%。

B. 假如高速缓存有 1024 字节，每 4 次读取（即访问情况）为：m h h h，循环读取，可知不命中率为 25%。

C. 在每一次引用 x[0][i] 和 x[1][i] 时，指向的缓存行是相同的，但由于每个组是有两个缓存行相联的（两路组相联），刚被缓存的 x[0][i] 并不会被替换，而

会在第二个缓存行中存入 $x[1][i]$ ，所以 m h h h 的循环不会发生改变，故不命中率为 25%。

D. 不会。在高速缓存是 512 字节，两路组相联，使用 LRU 替换策略时，高速缓存块大小为 16 字节的条件下，因为每个缓存行的第一次引用都是无法命中的，故不能降低不命中率

E. 会降低不命中率。增加了块的大小后，每一次缓存，有更多的内存能够进入缓存行，循环：m h h h h ... 将会越来越多（在一定范围内，块的大小越大，h 的数量越多），故不命中率会降低。

6.41

结果：25%

分析过程：

每一个 pixel 是长度为 4 个字节的结构体。由于高速缓存足够大，所以考虑每一个缓存行的情况。在每一次循环体中，第一次对 $buffer[i][j].r$ 的引用是不命中的 ($buffer[i][j].r = 0$)，随后缓存四个字节，故紧跟着的 g, b, a 三次引用都可以命中。与 6.34 的情况相似，在每一个缓存行中的命中情况为 m h h h，故不命中率为 $1/4 = 25\%$ 。