

# 计算机系统 HW 4

SA25011049 李宇哲

《深入理解计算机系统-第三版》书本第六章

6.25 6.29 6.34 6.38 6.41

## T1 (6.25)

\* 6.25 下面的表给出了一些不同的高速缓存的参数。对于每个高速缓存，填写出表中缺失的字段。记住  $m$  是物理地址的位数， $C$  是高速缓存大小(数据字节数)， $B$  是以字节为单位的块大小， $E$  是相联度， $S$  是高速缓存组数， $t$  是标记位数， $s$  是组索引位数，而  $b$  是块偏移位数。

高速缓存	$m$	$C$	$B$	$E$	$S$	$t$	$s$	$b$
1.	32	1024	4	4				
2.	32	1024	4	256				
3.	32	1024	8	1				
4.	32	1024	8	128				
5.	32	1024	32	1				
6.	32	1024	32	4				

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

## T2 (6.29)

\*\* 6.29 假设我们有一个具有如下属性的系统：

- 内存是字节寻址的。
- 内存访问是对 1 字节字的(而不是 4 字节字)。
- 地址宽 12 位。
- 高速缓存是两路组相联的( $E=2$ )，块大小为 4 字节( $B=4$ )，有 4 个组( $S=4$ )。

高速缓存的内容如下，所有的地址、标记和值都以十六进制表示：

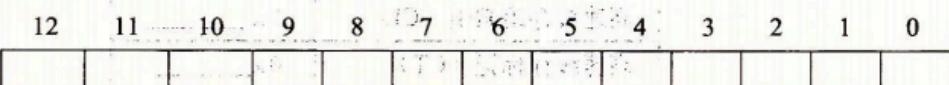
组索引	标记	有效位	字节0	字节1	字节2	字节3
0	00	1	40	41	42	43
1	83	1	FE	97	CC	D0
2	00	1	44	45	46	47
3	83	0	—	—	—	—
0	00	1	48	49	4A	4B
1	40	0	—	—	—	—
2	FF	1	9A	C0	03	FF
3	00	0	—	—	—	—

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



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	—	—
写	0x836	—	—
读	0xFFD	—	—

## A.

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

## B.

操作	地址	命中?	读出的值 (或者未知)
读	0x834	No	-
写	0x836	Yes	未知
读	0xFFD	Yes	0xC0

## T3 (6.34)

src 数组

c0	c1	c2	c3	
r0	m	m	h	m
r1	m	h	m	h
r2	m	m	h	m
r3	m	h	m	h

dst数组

c0	c1	c2	c3	
r0	m	m	m	m
r1	m	m	m	m
r2	m	m	m	m
r3	m	m	m	m

## T4 (6.38)

A.

$$4 * 16 * 16 = 1024$$

B.

`sizeof(point_color) == 16, B = 32`

```
1 | square[i][j].c = 0
```

miss, cache 2 point\_color

```
1 | square[i][j].m = 0
2 | square[i][j].y = 0
3 | square[i][j].k = 0
4 | square[i][j+1].c = 0
5 | square[i][j+1].m = 0
6 | square[i][j+1].y = 0
7 | square[i][j+1].k = 0
```

全部命中

因此 miss hit 总数为  $4 * 16 * 16 * 1/8 = 128$

C.

不命中率为  $1/8 = 12.5\%$

## T5 (6.41)

每次循环中  $buffer[i][j].r = 0$ , 因此永远会 miss,

$buffer[i][j].g = 0; buffer[i][j].b = 0; buffer[i][j].a = 0$ ; 永远命中

因此 miss rate 为  $1/4 = 25\%$