

第6章家庭作业

——201808010515 计科 1805 黄茂荣

****6.32** 假设程序使用作业 6.31 中的高速缓存，引用位于地址 0x0718 处的 1 字节字。用十六进制表示出它所访问的高速缓存条目，以及返回的高速缓存字节值。指明是否发生了高速缓存不命中。如果有高速缓存不命中，对于“返回的高速缓存字节”输入“—”。提示：注意那些有效位！

A. 地址格式（每个小框表示一位）：

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

B. 存储器引用：

参数	值
块偏移量 (CO)	0x_____
索引 (CI)	0x_____
高速缓存标记 (CT)	0x_____
高速缓存命中? (是 / 否)	
返回的高速缓存值	0x_____

解答

A 分析：假设是 6.31 中的高速缓存模型，则 $s=3$, $E=4$, $b=2$, 故第 0-1 位表示块偏移量，第 2-4 位表示组索引，5-12 位表示标记位。地址 0x0718 化为 16 进制表示为 11100011000，用题目中地址格式表示为

0	0	1	1	1	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

B 分析：CO, CI, CT 根据地址信息将其转换为相应的 16 进制数即可，判断高速缓存是否命中需要看有效位 V 有效 ($V=1$) 且标记位相同，对照 6.31 中的表格，找到组索引为 6 的组，发现第 4 行有效位为 1，且标记位为 0x38，和地址 0x0718 中标记位相同，故高速缓存命中。根据题意引用位于 0x0718 处的一字节，结合下表返回的高速缓存值为 FA。

4 路组相联高速缓存

索引	标记	V	字节 0~3	标记	V	字节 0~3	标记	V	字节 0~3	标记	V	字节 0~3
0	F0	1	ED 32 0A A2	8A	1	BF 80 1D FC	14	1	EF 09 86 2A	BC	0	25 44 6F 1A
1	BC	0	03 3E CD 38	A0	0	16 7B ED 5A	BC	1	8E 4C DF 18	E4	1	FB B7 12 02
2	BC	1	54 9E 1E FA	B6	1	DC 81 B2 14	00	0	B6 1F 7B 44	74	0	10 F5 B8 2E
3	BE	0	2F 7E 3D A8	C0	1	27 95 A4 74	C4	0	07 11 6B D8	BC	0	C7 B7 AF C2
4	7E	1	32 21 1C 2C	8A	1	22 C2 DC 34	BC	1	BA DD 37 D8	DC	0	E7 A2 39 BA
5	98	0	A9 76 2B EE	54	0	BC 91 D5 92	98	1	80 BA 9B F6	BC	1	48 16 81 0A
6	38	0	5D 4D F7 DA	BC	1	69 C2 8C 74	8A	1	A8 CE 7F DA	38	1	FA 93 EB 48
7	8A	1	04 2A 32 6A	9E	0	B1 86 56 0E	CC	1	96 30 47 F2	BC	1	F8 1D 42 30

参数	值
块偏移量 (CO)	0x0
索引 (CI)	0x6
高速缓存标记 (CT)	0x38
高速缓存命中?	是
返回的高速缓存值	0xFA

****6.33** 对于存储器地址 0x16EC 重复作业 6.32。

A. 地址格式（每个小框表示一位）：

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

B. 存储器引用：

参数	值
块偏移量 (CO)	0x_____
索引 (CI)	0x_____
高速缓存示记 (CT)	0x_____
高速缓存命中? (是/否)	
返回的高速缓存值	0x_____

解答

A 分析：假设是 6.31 中的高速缓存模型，则 $s=3$, $E=4$, $b=2$, 故第 0-1 位表示块偏移量，第 2-4 位表示组索引，5-12 位表示标记位。地址 0x16EC 化为 16 进制表示为 1011011101100，用题目中地址格式表示为

1	0	1	1	0	1	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

B 分析：CO, CI, CT 根据地址信息将其转换为相应的 16 进制数即可，判断高速缓存是否命中需要看有效位 V 有效 ($V=1$) 且标记位相同，对照 6.31 中的表格，找到组索引为 3 的组，发现当有效位为 1，发现高速缓存中标记位和地址 0x16EC 中标记位 B7 均不相同，故高速缓存不命中，返回值无。

标记位不匹配 4 路组相联高速缓存

索引	标记	V	字节 0~3	标记	V	字节 0~3	标记	V	字节 0~3	标记	V	字节 0~3
0	FO	1	ED 32 0A A2	8A	1	BF 80 1D FC	14	1	EF 09 86 2A	BC	0	25 44 6F 1A
1	BC	0	03 3E CD 38	A0	0	16 7B ED 5A	BC	1	8E 4C DF 18	E4	1	FB B7 12 02
2	BC	1	54 9E 1E FA	B6	1	DC 81 B2 14	00	0	B6 1F 7B 44	74	0	10 F5 B8 2E
3	BE	0	2F 7E 3D A8	CO	1	27 95 A4 74	C4	0	07 11 6B D8	BC	0	C7 B7 AF C2
4	7E	1	32 21 1C 2C	8A	1	22 C2 DC 34	BC	1	BA DD 37 D8	DC	0	E7 A2 39 BA
5	98	0	A9 76 2B EE	54	0	BC 91 D5 92	98	1	80 BA 9B F6	BC	1	48 16 81 0A
6	38	0	5D 4D F7 DA	BC	1	69 C2 8C 74	8A	1	A8 CE 7F DA	38	1	FA 93 EB 48
7	8A	1	04 2A 32 6A	9E	0	B1 86 56 0E	CC	1	96 30 47 F2	BC	1	F8 1D 42 30

参数	值
块偏移量 (CO)	0x0
索引 (CI)	0x3
高速缓存标记 (CT)	0xB7
高速缓存命中?	否
返回的高速缓存值	——

**** 6.37** 这道题测试你预测 C 语言代码的高速缓存行为的能力。对下面这段代码进行分析：

```
1      int x[2][256];
2      int i;
3      int sum = 0;
4
5      for (i = 0; i < 256; i++) {
6          sum += x[0][i] * x[1][i];
7      }
```

假设我们在下列条件下执行这段代码：

- `sizeof(int) == 4`。
 - 数组 `x` 从存储器地址 `0x0` 开始，按照行优先顺序存储。
 - 在下面每种情况中，高速缓存最开始时都是空的。
 - 唯一的存储器访问是对数组 `x` 的条目进行访问。其他所有的变量都存储在寄存器中。
- 给定这些假设，估计下列情况中不命中率。

A 情况 1: 假设高速缓存是 1024 字节，直接映射，高速缓存块的大小为 32 字节。不命中率是多少？

由 $C=B \times E \times S$, $C=1024$, $B=32$, 又因为是直接映射, 故 $E=1$, 所以可以得到 $S=32$ 。由于块的大小为 32 字节, 一个块中可以放 8 个 `int` 数, 又 $256 \times 8 = 1024$, 所以数组 `x[0][i]`, `x[1][i]` 对应相同的组即在同一块中 ($i=0, \dots, 255$), 在一次访问时, `x[0][i]`, `x[1][i]` 会交替加载, 驱逐, 导致全不命中, 故不命中率为 100%。

B 情况 2: 如果我们把高速缓存的大小翻倍到 2048 字节, 不命中率是多少？

当高速缓存大小翻倍后, 可以存放下整个数组 `x`, `x[0][i]`, `x[1][i]`, ($i=0, \dots, 255$) 每 8 个放在 cache 中的一个组, 每组只有一开始的冷不命中, 故不命中率为 $1/8=12.5\%$ 。

C 情况 3: 现在假设高速缓存是 1024 字节, 两路组相连, 使用 LRU 替换策略, 高速缓存块的大小为 32 字节, 不命中率是多少。

相对于情况 1, 采用两路组相连, 即 $E=2$, 虽然数组 `x[0][i]`, `x[1][i]` 对应相同的组中 ($i=0, \dots, 255$), 但是在一次访问时, 一个组中有两个块, 可以存放下 `x[0][i]` 和 `x[1][i]`。当发生冲突时, 利用 LRU 替换策略进行替换, 故不命中率和情况 2 一致, 为 12.5%。

D 对应情况 3, 更大的高速缓存的大小会帮助降低不命中率吗? 为什么能或者为什么不能?

不能降低。因为块大小不变时, 冷不命中的次数不可能被减小。

E 对应情况 3, 更大的块的大小会帮助降低不命中率吗? 为什么能或者为什么不能?

能降低。因为一个块的大小增加, 冷不命中的次数就会降低。

****6.42** 你正在编写一个新的 3D 游戏，希望能名利双收。你现在正在写一个函数，使得在画下一帧之前先清空屏幕缓冲区。你现在工作的屏幕是 640×480 像素数组。你工作的机器有一个 64KB 直接映射高速缓存，每行 4 个字节。你使用下面的 C 语言数据结构：

```
1 struct pixel {
2     char r;
3     char g;
4     char b;
5     char a;
6 };
7
8 struct pixel buffer[480][640];
9 int i, j;
10 char *cptr;
11 int *iptr;
```

有如下假设：

- $\text{sizeof}(\text{char}) == 1$ 和 $\text{sizeof}(\text{int}) == 4$ 。
- `buffer` 起始于存储器地址 0。
- 高速缓存初始为空。
- 唯一的存储器访问是对于 `buffer` 数组中元素的访问。变量 `i`、`j`、`cptr` 和 `iptr` 被存放在寄存器中。

下面代码中百分之多少的写会在高速缓存中不命中？

```
1     for (j = 0; j < 640; j++) {
2         for (i = 0; i < 480; i++){
3             buffer[i][j].r = 0;
4             buffer[i][j].g = 0;
5             buffer[i][j].b = 0;
6             buffer[i][j].a = 0;
7         }
8     }
```

解答：

由 $C = B * E * S$ ， $C = 64\text{kb} = 64 * 1024$ ， $B = 4$ ，又因为是直接映射，故 $E = 1$ ，所以可以得到 $S = 32 * 512$ 。每个结构体存放在一个块中。写入时，题中实现代码是按照列来写入，每次循环写入 4 次，第 1 次没有命中，其余 3 次写入时，均命中，故可以知道不命中率为 $1/4 = 25\%$ 。