

# 计算机组成原理

## Homework Solution 8

Made by TA

2023 年 6 月 15 日

注意：

1. 本次作业提交 DDL 为 6.7 下午 2:00 之前，超过该时间的提交会被扣除一定的作业分数。
2. 请特别注意按照题目所给的地址位宽完成题目。

**题目 1.** 现在一台电脑的处理器有**32 bits**的寻址空间，内存中的每个块的大小为**32 bytes**，电脑的 cache 能够存储 **16KIB** 数据。

1. 请问该电脑的 cache 能够存储多少个块？
2. 假设该电脑的 cache 采用直接映射的方式，那么地址中**TAG**、**Index**、**OFFSET**的位宽分别是多少？
3. 如果采用的是 4 路组相联的方式，那么**TAG**、**Index**、**OFFSET**的位宽分别是多少？

解答：

1.  $16\text{KIB}/32\text{B} = 2^{14}/2^5 = 2^9$
2. 18 bits, 9 bits, 5 bits
3. 20 bits, 7 bits, 5 bits

**题目 2.** 现在有一个 8bits 寻址空间的计算机，内存中的每个块的大小为 8 bytes，计算机的 cache 能够存储 32B 数据。cache 采用两路组相联的方式，采用 LRU 策略。现在有以下访问序列：

Address	Tag	Index	Offset	Hit/Miss/Replace	Miss Type
0b00000100					
0b00000101					
0b01101000					
0b11001000					
0b01101000					
0b11011101					
0b01001010					
0b00000100					
0b11001000					

本题一空 0.5 分最多扣 2 分。

**强制失效：**访问未被 cache 存储过的数据块。

**容量失效：**试图访问的数据块曾在 cache 中，但是因为 cache 容量不足，被替换掉了。（优先于冲突失效判断）

**冲突失效：**试图访问的数据块曾在 cache 中，cache 容量足够，但是因为组相联数不足，导致被替换掉了需要访问的数据块。

一种观点认为，冲突失效仅会发生在直接映射和组相联映射的 cache 中。确实存在一定的模糊性最后一个空填容量失效和冲突失效都未扣分。和老师讨论后，期末考试不会考失效类型判断相关问题（但可能考简答题）。

**解答：**

0b00000100	0000	0	100	M	强制失效
0b00000101	0000	0	101	H	
0b01101000	0110	1	000	M	强制失效
0b11001000	1100	1	000	M	强制失效
0b01101000	0110	1	000	H	
0b11011101	1101	1	101	R	强制失效
0b01000101	0100	0	101	M	强制失效
0b00000100	0000	0	100	H	
0b11001000	1100	1	000	R	容量失效

**题目 3.** 现在有一个 32 位字节寻址的 RISC-V 计算机，拥有 4 GiB 内存，一个 16 KiB 的 4 路组相联的 cache，每个 cache 块的大小为 32 byte 同时采用 LRU 策略。现在我们有如下的 C 代码, 假定第一次运行时 cache 为空：

```
#define SIZE_A 2048

typedef struct {
    int x; // int 为 32 bits
    int y[3];
} node;

int count(node *A, int x) { // 为简化问题，A 的首地址恰好位于一个块的首地址
    int k = 0;
    for (int i = 0; i < SIZE_A; i++) {
        if (A[i].x == x) {
            k++;
        }
    }
    return k;
}
```

在相同 A 在不同 x 下连续进行多次调用，回答下列问题：

1. 采用直接映射，对  $A[i].x$  的访问是否会产生 cache 失效？如果会，是哪一种失效？命中率是多少？
2. 如果采用 8 路组相联，对  $A[i].x$  的访问是否会产生 cache 失效？如果会，是哪一种失效？命中率是多少？
3. 如果采用全相联映射且将 cache 的策略改为 MRU，对  $A[i].x$  的访问是否会产生 cache 失效？如果会，是哪一种失效？命中率大约是多少？
4. cache 容量满后，出现的失效一定为容量失效吗？如果不是，举例说明。
5. 通过增加块的数量来增加 cache 容量一定能提高命中率吗？如果不是，举例说明。

解答:

由于失效定义的模糊性，1,2 只要答了强制失效都没扣分，第三问失效类型答案唯一（全相联没有冲突失效）。

1. 会产生强制失效和容量失效，一共能存储 9bit 块但是需要 cache 大约  $2^{11}/2 = 2^{10}$  个块，命中率为 0.5，因为一个 node 有 4 个 int，也即 16B，从而一个 cache 行能缓存两个 node。例如访问  $A[0].x$  出现强制失效，访问  $A[1].x$  命中，访问  $A[2].x$  出现强制失效，访问  $A[3].x$  命中，以此类推。
2. 答案同上
3. 会产生强制失效和容量失效，第一次运行时命中率为 0.5，第一次结束后大约有  $2^9$  块留在 cache 中并且由于采用 MRU 策略留下的是前一半的块，第二次运行时前一半命中率为 1，后一半命中率为 0.5，因此命中率为 0.75，此后均与第二次运行类似，故充分多次运行后命中率大约为 0.75
4. 不一定，仍然会出现强制失效，例如此时访问一个从未被访问过的块，那么对它的访问则仍是强制失效。
5. 不一定，对于本题的例子，若该循环只出现一次，且 node 结构体包含 8 个 int，则循环每次对  $A[i].x$  都会产生强制失效。cache 命中率有时还依赖于程序自身的性质。