

计算机体系架构 第八周作业

范云潜 18373486

微电子学院 184111 班

日期: 2020 年 11 月 24 日

作业内容: 7.1, 7.3, 7.6, 7.11, 7.25, 7.32, 7.36。

Problem 7.1

如果不考虑成本的话, 更大的 SRAM 会占据更大额度片上面积, 进而影响其他的模块, 并且缺少 RAM 接口, 难以兼容以 RAM 接口为主的 DMA 模块, 速度成为瓶颈。

Problem 7.3

对数据来说, 时间连续, 但是空间性不强, 哈希表是一个很好的例子。

```
Get: HashMap map
data = map[key]
for op in {find_index, find_value, ...}:
    op.answer = op(data)
```

Problem 7.6

对指令来说, 时间和空间性都不强, 也就是指令的跳跃性很强。

```
Get: UserInput keyboard
data, choice = read(keyboard)
case(choice):
    type_1:
        multiply(data[0], data[1])
    type_2:
        add(*((int*)data[2]), 0xff3e)
    type_3:
        copy(from=data, to=new_data)
```

Problem 7.11

在不进行优化的情况下, 通过时钟函数进行计时, 代码如下, y 优化比例大概有 2.5 倍。

```
#include <stdio.h>
#include <time.h>

int main()
{
    static int array[10000][100000];
    clock_t start, end;
    start = clock();
```

```

double t1, t2;
for (int i = 0; i < 10000; ++i)
{
    for (int j = 0; j < 100000; ++j)
    {
        array[i][j] = 2 * array[i][j];
    }
}
end = clock();
t1 = end - start;
printf("the cost of for-row is %d\n", end - start);

start = clock();
for (int i = 0; i < 100000; ++i)
{
    for (int j = 0; j < 10000; ++j)
    {
        array[j][i] = 2 * array[j][i];
    }
}
end = clock();

t2 = end - start;
printf("the cost of for-col is %d\n", end - start);

printf("the ratio is %f", t1 / t2);

return 0;
}

>>> the cost of for-row is 7779271
>>> the cost of for-col is 19284270
>>> the ratio is 0.403400

```

Problem 7.25

```

addr:
2:  empty and load to way1(1-8)
3:  hit
11: empty and load to way2(9-16)
16: hit
21: miss and load to way1(17-24)
13: hit
64: miss and load to way1(57-64)
48: miss and load to way2(41-48)
19: miss and load to way1(17-24)
11: miss and load to way2(9-16)
3:  miss and load to way1(1-8)

```

```

22: miss and load to way2(17-24)
4:  hit
27: miss and load to way2(25-32)
6:  hit
11: miss and load to way2(9-16)
buffer:
way1: 1   2   3   4   5   6   7   8
way2: 9   10  11  12  13  14  15  16

```

Problem 7.32

需要计算整体未命中的惩罚: $Total = I_{penalty} + D_{penalty}$ 。

因此分别为:

$$t1 = (6 + 1) \times (4\% + 6\%/2) = 0.49$$

$$t2 = (6 + 4) \times (2\% + 4\%/2) = 0.4$$

$$t3 = (6 + 4) \times (2\% + 3\%/2) = 0.35$$

Problem 7.36

18 32K x 16-bit

每个数据是 4 * 16-bit, 偏移为 2 位, 那么共有 18 * 8 K 个数据, 每组 3 路, 共 48 K 组, 那么组索引需要 $\log_2 64K = 16$ 位, 因此标记有 14 位。

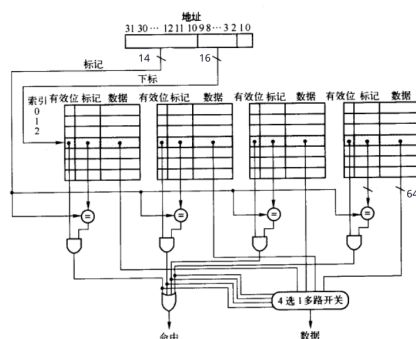


图 1: 7.36