

作业 1

1. 假定执行某种指令 I 需要 20 个时钟周期, 该指令在程序中出现的频度为 10%, 其他所有指令的平均 CPI 为 5, 则 CPU 执行指令 I 所用时间占整个 CPU 时间的百分比是多少? 如果通过对硬件进行改进, 能使指令 I 的执行时间缩短为 10 个执行周期, 但同时会使 CPU 时钟周期延长 10%, 你认为是否应该采取这种措施?

2. 若机器 M1 和 M2 具有相同的指令集, 其时钟频率分别为 0.8GHz 和 1.6GHz。在指令集中有 5 种不同类型的指令 A~E。下表给出了在 M1 和 M2 上每类指令的平均时钟周期数 CPI

M1 和 M2 的 CPI 以及三个程序的指令频度

指令类	A	B	C	D	E
M1	1	2	2	3	4
M2	2	2	4	5	6

请回答下列问题:

(1) M1 和 M2 的峰值 MIPS 各是多少?

(2) 假定某程序 P 的指令序列中, 5 类指令具有完全相同的指令条数, 则程序 P 在 M1 和 M2 上运行时, 哪台机器更快? 快多少? 在 M1 和 M2 上执行程序 P 时的平均时钟周期数 CPI 各是多少?

3. 某 C 语言程序, 从终端输入 float 型数值: 61.419997, 将其打印输出, 得: 61.419998。当输入分别为以下数值时: 61.42, 61.49, 62.335, 请问打印输出为多少?

4. 考虑下列 C 语言程序代码:

```
int i = 65535;
short si = (short) i;
int j = si;
```

假定上述程序段在某 32 位机器上执行, sizeof(int)=4, 则变量 i、si 和 j 的值分别是多少? 为什么?

5. 假设某字长为 8 位的计算机中, 带符号整数采用补码表示, $x=-68$, $y=-80$, x 和 y 分别存放在寄存器 A 和 B 中。请回答下列问题 (要求最终用十六进制表示二进制序列)

(1) 寄存器 A 和 B 中的内容分别是什么?

(2) 若 x 和 y 相加后的结果存放在寄存器 C 中, 则寄存器 C 中的内容是什么? 运算结果是否正确? 加法器最高位的进位 Cout 是什么? 溢出标志 OF、符号标志 SF 和零标志 ZF 各是什么?

(3) 若 x 和 y 相减后的结果存放在寄存器 D 中, 则寄存器 D 中的内容是什么? 运算结果是否正确? 加法器最高位的进位 Cout 是什么? 溢出标志 OF、符号标志 SF 和零标志 ZF 各是什么?

(4) 对于带符号整数的减法运算, 能否直接根据 CF 的值对两个带符号整数的大小进行比较?

6. 已知 C 语言中的按位异或运算 “XOR” 用符号 “^” 表示。对于任意一个位序列 a 存在

$a \oplus a = 0$ 。C 语言程序可以利用这个特性来实现两个数值交换的功能。以下是一个实现该功能的 C 语言函数：

```
void xor_swap(int *x, int *y)
{
    *y = *x ^ *y; /* 第一步 */
    *x = *x ^ *y; /* 第二步 */
    *y = *x ^ *y; /* 第三步 */
}
```

假定执行该函数时 $*x$ 和 $*y$ 的初始值分别为 a 和 b ，即 $*x=a$ 且 $*y=b$ ，请说明每一步执行结束后 x 和 y 各自指向的内存单元中的内容分别是什么？

7. 假定某个实现数组元素倒置的函数 `reverse_array` 调用了第 6 题中给出的 `xor_swap` 函数：

```
void reverse_array(int a[], int len)
{
    int left, right = len - 1;
    for (left=0; left <= right; left++, right--)
        xor_swap(&a[left], &a[right]);
}
```

当 `len` 为偶数时，`reverse_array` 函数的执行没有问题。但是，当 `len` 为奇数时，函数的执行结果不正确。请问当 `len` 为奇数时会出现什么问题？最后一次循环中的 `left` 和 `right` 各取什么值？最后一次循环中调用 `xor_swap` 函数后的返回值是什么？对 `reverse_array` 函数做怎样的改动就可消除该问题？

提交作业

作业（word 或 pdf 格式）拷贝到一个文件夹中，命名为：

作业 n

其中， $n=1\cdots 3$ 为第 n 次作业

课程结束时，将 3 次作业与 6 次实验拷贝到同一个文件夹中，命名为如下格式：

班号-学号-姓名

以班为单位一起提交。