PA-1 实验报告

陈奕诺 191220013

2020.10.5

一、实验目的

1.学习计算机系统数据的表示和存取的原理

2.学习整数加减乘除等基础运算在计算机中的表示方式和实现方式

3.在整数运算的基础上学习浮点数加减乘除运算在计算机中的表示方式和实现方式

4.完成 ALU 和 FPU 的构建

二、实验原理

1.计算机中整数的表示和运算（主要是 eflag 的判断，具体在 i386 手册中有明确规定）（重点在整数补码运算）

2.计算机中浮点数的运算，中间结果的获取，中间结果的规格化。（重点在结果的规格化）

三、实验环境

Linux 平台

四、实验内容

PA1-1 按照要求完成数据结构的更正，了解数据的表示和存取，熟悉平台操作流程

PA1-2 按照要求完成整数加减乘除运算代码的补充，判断相应的 eflags，了解整数加减乘除等基础运算在计算机中的表示方式和实现方式

PA1-3 按照要求完成浮点数数加减乘除运算代码的补充，并完成相应的规格化要求，了解浮点数加减乘除等基础运算在计算机中的表示方式和实现方式。学习浮点数运算中的各种特殊情况处理。

五、实验结果

其中在计算浮点数的乘法过程中遇到了较大的困难，但最终通过查看 b 站上现有的教学视频，了解到了通过改变阶码来实现小数点的移动，使其始终在传入规格化函数之时，小数点始终处在第 26 位的位置，最终在注释和课本的帮助下完成了此次 PA 的任务。

实现了既定的实验目标。

六、思考题

1.C 语言中的 struct 和 union 关键字都是什么含义，寄存器结构体的参考实现为什么把部分 struct 改成了 union?

struct 和 union 都是一种特殊的数据结构类型，struct 是结构体，union 是联合体，都能将一系列不同类型的数据整合在一起，区别是 union 类型存储时，不同的数据可能存储在同一位置，而 struct 则不会。

寄存器结构体的参考实现中部分 struct 改为 union 是为了保证存储的数据的一致性，避免因为存储位置不同导致的两个不同类型的数据不符合对应关系的情况出现。

2.为浮点数加法和乘法各找两个例子：

1）对应输入是规格化或非规格化数，而输出产生了阶码上溢结果为正（负）无穷的情况；
2） 对应输入是规格化或非规格化数，而输出产生了阶码下溢结果为正 （负）零的情况。是否都能找到？若找不到，说出理由

加法:

（1）输入两个这样的浮点数：符号 0 阶码 254 尾数任意

如 0 11111110 11111111111111111111111

（2）没有这样的例子，由于计算浮点数的过程中，我们计算中间数时是通过小阶向大阶看起齐的方式来实现对阶操作的，也就是计算的得到的中间数，其阶码一定在范围内，而不会小于 0。接下来如果在规格化阶段没有出现阶码下溢，就不会有相应的情况。但是我们可以看到我们在规格化操作中若出现左规的情况，其判断条件为 while (((sig_grs >> (23 + 3)) == 0) && exp > 0)也就是 exp 最小为 0 就会停止左规，而此时的数为非规格化数，尾数右移一位后得到非规格化数，阶码为 0，并不会出现阶码下溢的情况，此时如果尾数为 0 则结果为 0，但这是舍入的结果与下溢的含义不同。

乘法:

（1）输入两个这样的浮点数：符号 0 阶码 254 尾数任意

如 0 11111110 11111111111111111111111

（2）输入两个这样的浮点数：符号 0 阶码 1 尾数任意

如 0 00000001 11100000000000000000000