

# PA1 实验报告

161220096 欧阳鸿荣

## 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？

C 语言中的 struct 关键字含义是结构体类型，而 union 关键字是联合体类型。

联合体的各个成员共用内存，并应该同时只能有一个成员得到这块内存的使用权（即对内存的读写），而结构体各个成员各自拥有内存，各自使用互不干涉。所以，寄存器中某种意义上来说，联合体比结构体节约内存。而在寄存器结构体的参考实现中，把部分在实际使用情况下不会被覆盖的结构体改成联合体有利于节约内存，并且使得首地址保持一致便于访问。

## 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。

浮点数加法：

1.800000H+7F000000H=7F800000H    80800000H+7F000000H=FF800000H

2.在浮点数加法运算中，有一步名为对阶的操作，在这个步骤中，向高阶对齐，因此不会发生阶码下溢的情况

浮点数乘法：

1.3FC00000H\*2H=7F800000H    7FC00000H\*2H=FF800000H

2.F000000H\*F000000H=00000000H    40000000H\*2H=80000000H

## 3.实验中遇到的瓶颈与困难：

- 1) 对于 OF 和 CF 标志位理解不深入，尤其是 CF 在减法中的借位
- 2) 对于 sub 和 sbb 这两个运算，最开始的版本过不了 0x80000000 这个测试样例  
后来经过细想以及与同学的讨论才得到正确的做法

```
//OF
uint32_t diff=temp2+~temp1s+1;
uint32_t temp2_neg=temp2>>31;
uint32_t temp1_neg=temp1s>>31;
uint32_t d_neg=diff>>31;
cpu.eflags.OF=(~(temp2_neg^~temp1_neg)&(temp2_neg^d_neg));
```

- 3) 一开始对于粘位的理解错误
- 4) 对于取低三位，应为 temp&0x7，误写成 temp&0x111，debug 了好一阵子  
对于二进制和 16 进制的切换有时候还是会乱