

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:** 用于表示固定多个、类型可以不同的元素所构成的符合数据,元素在逻辑上没有次序关系。**union:** 成员相互互斥,不能同时使用它们,会分阶段地把它作为不同类型来使用。**struct** 的成员是连续占用空间,改变其中的比如 `AL` 并不能影响 `AX` 和 `EAX`,因此不符合要求。而 **union** 的成员示则共享一个 32 位的空间,那么可以做到统一的改变,符合寄存器的要求。

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。

(1) `cb189680 + 7f800000 = 7f800000(inf)`

`cb189680 + ff800000 = ff800000(-inf)`

`be4ccccd * ff800000 = 7f800000(inf)`

`be4ccccd * 7f800000 = ff800000(-inf)`

(2)加法没有阶码下溢

因为加法的阶码就是 2 个加数解码中较大的阶码,左右规也不能使其产生下溢。

`be4ccccd * 80400000 = cccd(0)`

`be4ccccd * 800000 = 8019999a(-0)`