

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？

结构类型 (struct) 用于表示由固定多个类型可以不同的元素所构成的复合数据，这些元素之间在逻辑上没有次序关系。

联合类型 (union) 的成员是互斥的，在程序中不能同时使用它们。对于一个联合类型的变量，在程序中将会分阶段地把它作为不同的类型来使用。

结构类型的变量在内存中占用一块连续的存储空间，其各个元素依据其在结构类型中的定义次序存储在这块类型空间中；联合类型的所有成员占用同一块内存空间，该内存空间的大小为其最大成员所需要的内存空间的大小。寄存器结构体所表现的是，从机器层面来说，模拟寄存器的内存空间是叠加的而不是并列的，存放在模拟寄存器中的数据存放在一块而不是多块内存空间中，只是在不同的访问和调用方式下被解读为不同的数据类型。

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；

加法： $3e+38 + 3e+38 = \text{inf}$ ， $-3e+38 + (-9e+37) = -\text{inf}$ ；

乘法： $1e+20 * 1e+20 = \text{inf}$ ， $3e+38 * (-1.5) = -\text{inf}$ ；

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：不能找到。若输入两个数符号相同，则和的绝对值比两个加数都大，不可能发生阶码下溢，若输入两数符号相反，由于非规格化数表示的最小绝对值固定，次小绝对值为这个值的两倍，这两个数相减不会阶码下溢，两个相同的数相减结果直接为零，也不发生下溢；

乘法： $-1e-30 * (-1e-20) = 0$ ， $1.4013e-45 * (-0.1) = -0$ 。