

## 1-1

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？

union 是特殊的 class 类型，它在一个时刻只能保有其一个非静态数据成员。

union 的多个成员 共享同一个存储空间，使得我们可以用不同的方式访问 reg

（在其他一些情况下还可以节省空间等）

（但使用 union 时要保证不同时访问 union 的不同成员）

而 struct 的各个成员的存储空间互相独立

## 1-2

写 pa1-2 时遇到的问题：

1. 开了两个桌面 两个代码文件 写了的一直没存进去..
2. 想清楚有效位，分清哪里要 sign\_ext 哪里要  $(0xffffffff \gg (32 - data\_size))$  等
3. 写 shl 的时候理解错了（其实不管这句话就好了）好像并不能只改变低 data\_size 位..

*特别说明：针对上面四个移位操作，约定只影响 dest 操作数的低 data\_size 位，而不影响其高  $32 - data\_size$  位。标志位的设置根据结果的低 data\_size 位来设置。*

## 1-3

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。

阶码上溢：

加法      7F7FFFFF（最大的规格化数）+ 任何正数  
乘法      7F7FFFFF \* 任何正数  
            或 7F000000 + 7F000000 ( $2^{127} + 2^{127}$ )  
            7F000000 \* 7F000000 ( $2^{127} * 2^{127}$ ) 等等

阶码下溢：

加法      不会出现阶码下溢 能表示出来的数做加法仍能表示出来  
乘法      00800000 \* 00800000 ( $1 * 2^{(-126)} * 1 * 2^{(-126)}$ )  
            00400000 \* 00400000 ( $0.1 * 2^{(-126)} * 0.1 * 2^{(-126)}$ ) 等等

写 pa 1-3 中遇到的问题：

1. 阶码上溢和下溢时对 exp 的处理，应该分别设为 0xff 和 0

```
if (exp >= 0xff) //阶码上溢
{
    /* TODO: assign the number to infinity */
    exp = 0xff;
    sig_grs = 0;
    overflow = true;
}

if (exp < 0) //阶码下溢
{
    /* TODO: assign the number to zero */
    exp = 0;
    sig_grs = 0;
    overflow = true;
}
```

2. 去掉保护位时破坏规格化情况的处理：  
应该考虑到两种情况：规格化数和非规格化数

```
if((sig_grs >> 23) > 1) {  
    sig_grs = sig_grs >> 1;  
    exp = exp + 1;  
    if(exp >= 0xff){  
        exp = 0xff;  
        sig_grs = 0;  
        overflow = true;  
    }  
}  
else if( ((sig_grs>>23) == 1) && exp == 0){  
    exp++;  
}
```