

PA1实验报告

PA1-1

- 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` 为结构体类型，为其中的每一个成员变量分配各自的单独的空间；`union` 为联合类型，根据其中成员变量中所占最大空间的成员分配空间，所有成员共享同一片内存空间。
-- 寄存器结构体的实现将 `struct` 改成了 `union` 实现了不同变量名称对同一片内存空间不同数据位段的访问，更符合实际寄存器的实现、更便于数据的存取。

PA1-2

- ADC需要结合CF的取值判断
考虑 $CF == 1$ 时， $src == res$ 的情况是否产生进位？
产生进位
- SUB为什么不能对减数取补码后，简单复用add的CF判断标准？
计算机的补码在模运算系统下可保证结果的正确性，故若减数为正数、被减数减小时，一定会产生ADD时的进位情况。

PA1-3

- 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。
注:仅考虑IEEE754标准下单精度浮点数

1. 加法例子：

- `0 11111110 11111111111111111111111111111111 + 0 11111110 11111111111111111111111111111111`
两规格化数相加，结果超出单精度浮点数所表示最大范围，结果设置为正无穷产生阶码上溢。
`0 00000000 11111111111111111111111111111111 + 0 11111110 11111111111111111111111111111111`
规格化数与非规格化数相加，也产生阶码上溢，结果为正无穷。
- `0 00000001 00000000000000000000000000000001 + 1 00000001 00000000000000000000000000000001`
两规格化数相加，发生下溢，结果为正零
不存在两数相加结果发生下溢，结果为负零情况：两数相同时相加为正零，且加法中不可能出现精度超出表示范围的加数。

1. 乘法例子：

- `0 11111110 11111111111111111111111111111111 * 0 11111110 11111111111111111111111111111111`
两规格化数相乘，结果超出单精度浮点数所表示最大范围，结果设置为正无穷产生阶码上溢。
`1 11111110 11111111111111111111111111111111 * 0 11111110 11111111111111111111111111111111`
两规格化数相乘，结果超出单精度浮点数所表示最大范围，结果设置为负无穷产生阶码上溢。
- `0 00000000 00000000000000000000000000000001 * 0 00000000 00000000000000000000000000000001`
两小正数相乘，结果超出范围产生下溢，结果为正零。
`1 00000000 00000000000000000000000000000001 * 0 00000000 00000000000000000000000000000001`
小正数与小负数相乘，结果超出范围产生下溢，结果为负零。

231220107 康峻赫