FP16\_MUX算法说明

该模块用于FP16规格化数的相乘，输出为FP16规格化数

1. Step1：check

这一阶段需要判断输入的合理性，0，正常，溢出

* 阶数和符号正常分离
* 如果有一个数的阶数是11111，则溢出，标志位置1
* 如果有一个阶数和尾数都是0（00000\_0000000000），则是0，取一个记录为0
* 如果正常，拼接{1，最低10位}作为尾数

1. Step2：calculate

这一阶段初步计算符号位，尾数和阶数

* 尾数相乘，最多22位
* 阶数相加再减15
* 符号位异或

1. Step3：carry

这一阶段根据规格化要求，进行第一次进位处理

* 如果尾数的第22位是1，则需要进位，尾数右移一位，阶数+1
* 如果阶数本身是30，+1则溢出；小于30，+1不会溢出

1. Step4：round to nearest even

这一阶段需要根据舍弃部分情况决定舍入

* 舍入原则是舍入到最近可以表示的数，如果距离一样，则偏好偶数。具体算法为[被舍弃部分最高位 && （保留部分最低位 || 被舍弃部分次高位及更低位是否有1）]
* 尾数保留部分是第20-11位
* 如果判断为真，则保留部分再+1，假则保留部分不变

1. Step5：carry again

由于step4中的入操作可能导致再次进位，这一阶段处理第二次进位

* 如果尾数的第12位为1则需要进位，尾数右移一位，阶数+1
* 如果阶数本身大于等于30，则+1会溢出；小于30，则+1不会溢出，照常

1. Step6：result

这一阶段可以得出结果

* 如果标志位overflow为真，或者阶数第七六位是01，则溢出，拼接{符号，0x7fff}
* 其他如果阶数是0或者阶数的第七位是1，则下溢出，输出0x0000
* 其他正常，拼接{符号位，阶数低五位，尾数低十位}输出