FP16\_MUX算法说明

# step1：calculate

这一阶段初步计算符号位，尾数和阶数

## 两个输入的[9:0]位**(& 0x03ff)**，在[10]位拼接被省略的1**(| 0x0400)**，再相乘得rmcache

## 两个输入的[14:10]位**(>> 10) & 0x001f)**，相加再减15得expcache

## 两个输入的[15]位**(>>15)**，相与得sign

# step2：overflow and carry

这一阶段进行第一次溢出处理

## 再step1尾数计算过程中，是1.xxxxxxxxxx乘1.xxxxxxxxxx，两者都是十一位数(小数点后十位)，计算结果最多二十二位数，其中低二十位为小数点后，小数点位于第二十位和二十一位中间，二十一位是小数点前的第一位。根据FP16归约化规则，若第二十二位为1，则需要多计出一阶数,同时小数点左移一位

## 如果rmcache[21]为1**(rmcache & 0x00200000) == 0x00200000)**则判断为溢出

## 尾数右移一位等效小数点左移一位

## 阶数加一

## 尾数右移一位会把最低位顶掉，用lastbit保存下来

# step3：round to nearest even

这一阶段进行舍入处理

## step2中已经说明了尾数小数点后有20位，但FP16只能保存10位，需要对尾数进行舍入处理。舍入原则为舍入到最近的可以表示的数，如果距离一样，则偏好偶数

## 入的判断条件为[rmcache[9]&& (rmcache[10] || rmcache低八位是否有1)]，假则舍，不作操作

## 真则入，rmcache[10]+1

# step4：overflow and carry again

step3中的入操作可能导致再次溢出，这一阶段处理第二次溢出

## 如果rmcache[21]为1**(rmcache & 0x00200000) == 0x00200000)**则判断为溢出

## 尾数右移一位等效小数点左移一位

## 阶数加一

# step5：result

这一阶段就可以计算结果

## FP16只有五位用于存储阶数，范围0-31

## 若阶数大于31，则上溢出，符号位不变，其余位置1进行溢出保护

## 若阶数处在0-31范围内，正常，拼接符号位、阶数低五位**(expcache & 0x001f)**、尾数[19:10]**(rmcache & 0x000ffc00)**作为输出

## 若阶数小于0，则下溢出，符号位不变，其余位置0进行溢出保护