# 浮点数的表示和计算

2019年4月12日 0:50

## 一、浮点数的表示：

single: 8 bits　　single: 23 bits

double: 11 bits　　double: 52 bits

| S | Exponent (yyyy+Bias) | Fraction (xxxx) |
|---|---|---|

$$x = (-1)^S \times (1 + Fraction) \times 2^{(Exponent - Bias)}$$

## 二、浮点数的范围：

Ensures exponent is unsigned

Single: Bias = 127; Double: Bias = 1203

Smallest value of single：

$\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$

Largest value of double：

$\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$

## 三、浮点数的精度：

Relative precision：

$\Delta A/|A| = 2^{-23} \times 2^{exponent} / |1 \times 2^{exponent}| = 2^{-23}$

ulp：

One-half ulp

## 四、浮点数运算：

加法：

1.align　指数看齐大的，右移较小的

2.add

3.normalization &　check overflow

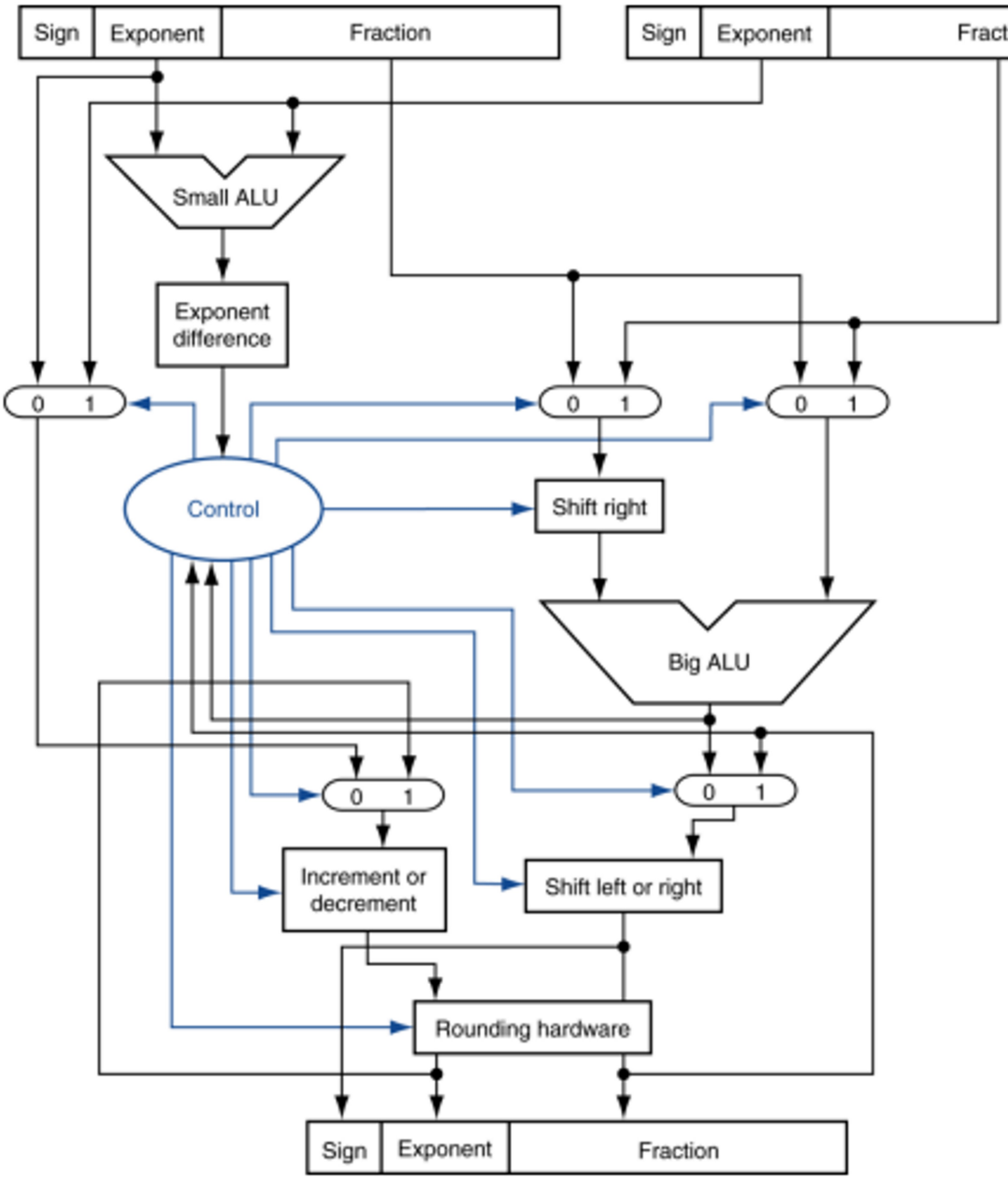
4.round & renormalization

加法器硬件：

·比整数adder复杂很多，耗时较长，放在一周期内会显著增大时钟周期，因此占用多个cycle

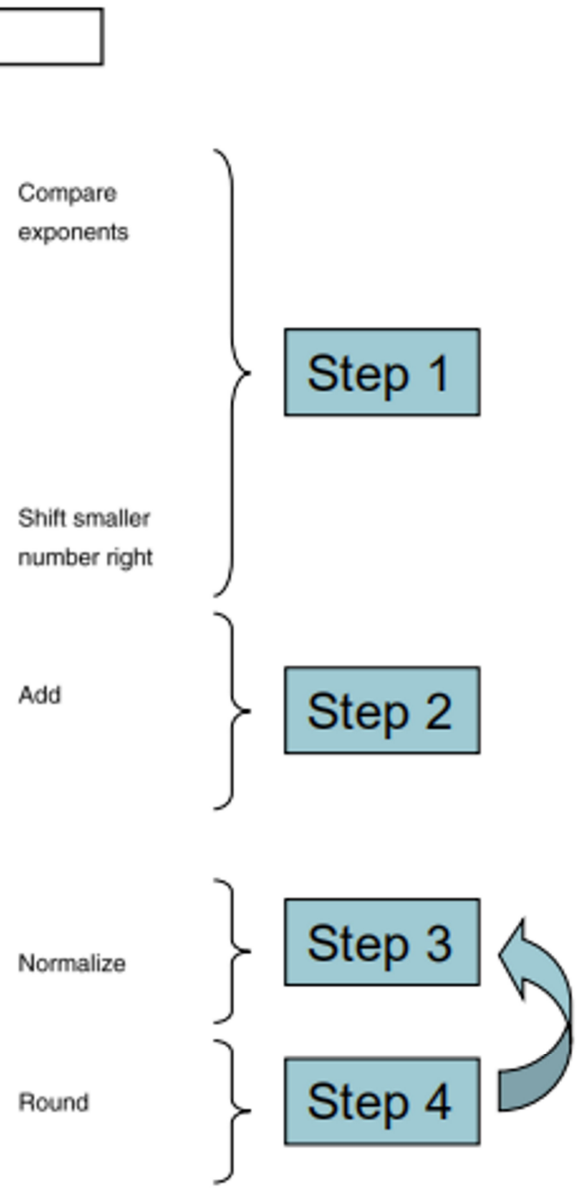
·可以使用pipeline

# FP Adder Hardware




【

1. 5个（0，1）选择器
2. 内容包含五个部分：比较指数，右移较小的具体数，加，归一化，round
3. 元件从上到下依次：
   比较指数 small ALU,exponent difference
   偏移小的 shift right
   加 Big ALU
   归一化 Increment or decrement,shift left or right
   Round Rounding hardware
4. ADD出的结果和round出的具体数会连接到control


Compare
exponents
Step 1
Shift smaller
number right
Add
Step 2
Normalize
Step 3
Round
Step 4

5. control链接所有的shift和（0，1）和归一化和round部分的所有元件
6. Shift left or right的结果会给定结果的sign

问题：凭什么可以6.

浮点数硬件运算器：加减乘除倒数，开方，整数和浮点数的转换。常花费不仅一个周期，但可以pipeline并行

五、浮点数指令

Ldc1 sdc1 lwc1 swc1　c.lt/le/eq.s　bc1t/bc1f

浮点数操作指令只操作浮点寄存器（cop1）

注意双精度只能对偶数编号浮点寄存器操作

六、进位模式

1.有模式可供选择

2.nearest even：　GRS

FP运算也有SIMD