Chapter 03: Computer Arithmetic

Lesson 09:

Arithmetic using floating point numbers

Objective

- To understand arithmetic operations in case of floating point numbers

Multiplication of Floating Point Numbers

Step 1

- IEEE floating-point numbers use a biased representation exponents
- Add the exponent fields two floating-point numbers
- Treat exponent fields as integers for addition
- Subtract the bias value from the result

Step 2

- Multiply the mantissas of the two numbers
- Using techniques analogous to those used to multiply decimal numbers, and to add their exponents

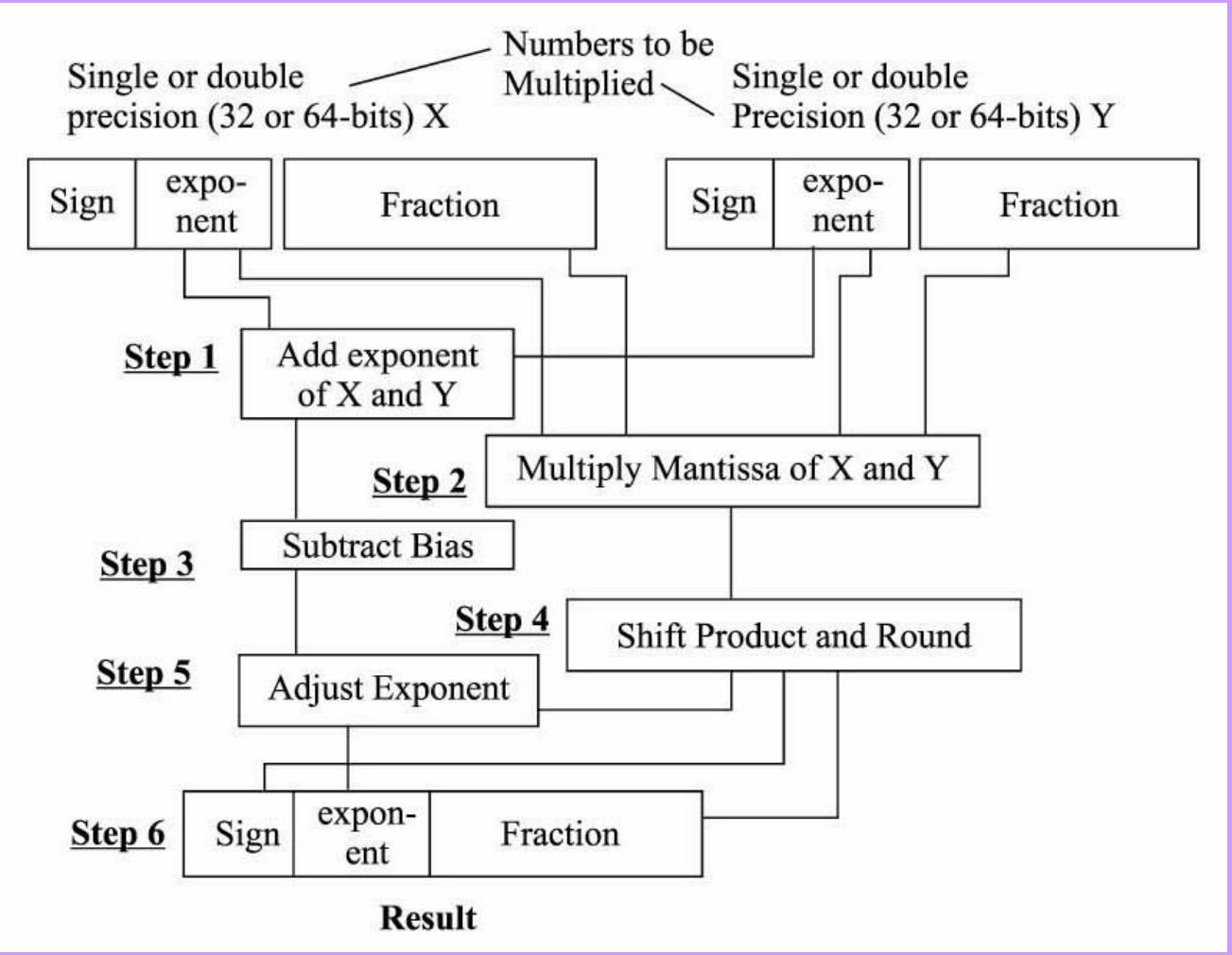
Step 3

- Multiplied mantissas─ the result may need to be shifted so that only 1 bit remains to the left of the binary point (i.e., so that it fits the form $1.xxxxx_2$, and the value of the sum of exponents incremented
- Shift such that the value of the mantissa $\times$ $2^{exponent}$ remains the same

Step 4

- The product of the mantissas after shifting— may also have to be rounded to fit within the number of bits allocated to the fraction field, since the product of two *n*-bit mantissas may require up to 2× *n* bits to represent exactly
- Shifted and rounded mantissa — Assemble final product out of the product of the mantissas and sum of exponents

Floating- Point Multiplication



Multiply 2.5 by 0.75 Using single-precision floating-point numbers

- 2.5 = 0b0100 0000 0010 0000 0000 0000 0000 0000 (
- Exponent field of 0b10000000,
- Fraction field of 0b0100000 00000000 0000 0000
- Mantissa of $1.010\ 0000\ 0000\ 0000\ 0000\ 0000_2$

Multiply 2.5 by 0.75 Using single-precision floating-point numbers

- 0.75 = 0b0011 1111 0100 0000 0000 0000 0000 0000
- Exponent field of 0b01111110
- Fraction field of 0b100 0000 0000 0000 0000 0000
- Mantissa of $1.100\ 0000\ 0000\ 0000\ 0000\ 0000_2$

Multiply 2.5 by 0.75

- Adding the exponent fields directly and subtracting the bias gives a result = 0b01111111
- Multiplying the mantissas gives a result = 1.111 0000 0000 0000 0000 0000_2
- Converts into a fraction field of 0b111 0000 0000 0000 0000 0000
- Result is 0b0011 11111111 0000 0000 0000 0000 0000 = $1.111_2 \times 2^0 = 1.875$

Division of Floating Point Numbers

Floating-point division

- Very similar to multiplication– the hardware computes the quotient of the mantissas and the difference between the exponents of the numbers being divided, adding the bias value to the difference between the exponent fields of the two numbers to get the correct biased representation of the result
- The quotient of the mantissas then shifted and rounded to fit within the fraction field of the result

Addition of Floating Point Numbers

Floating-point addition

- As with adding numbers in scientific notation, the first step is to shift one of the inputs until both inputs have the same exponent
- In adding floating-point numbers, the number with the smaller exponent is right-shifted

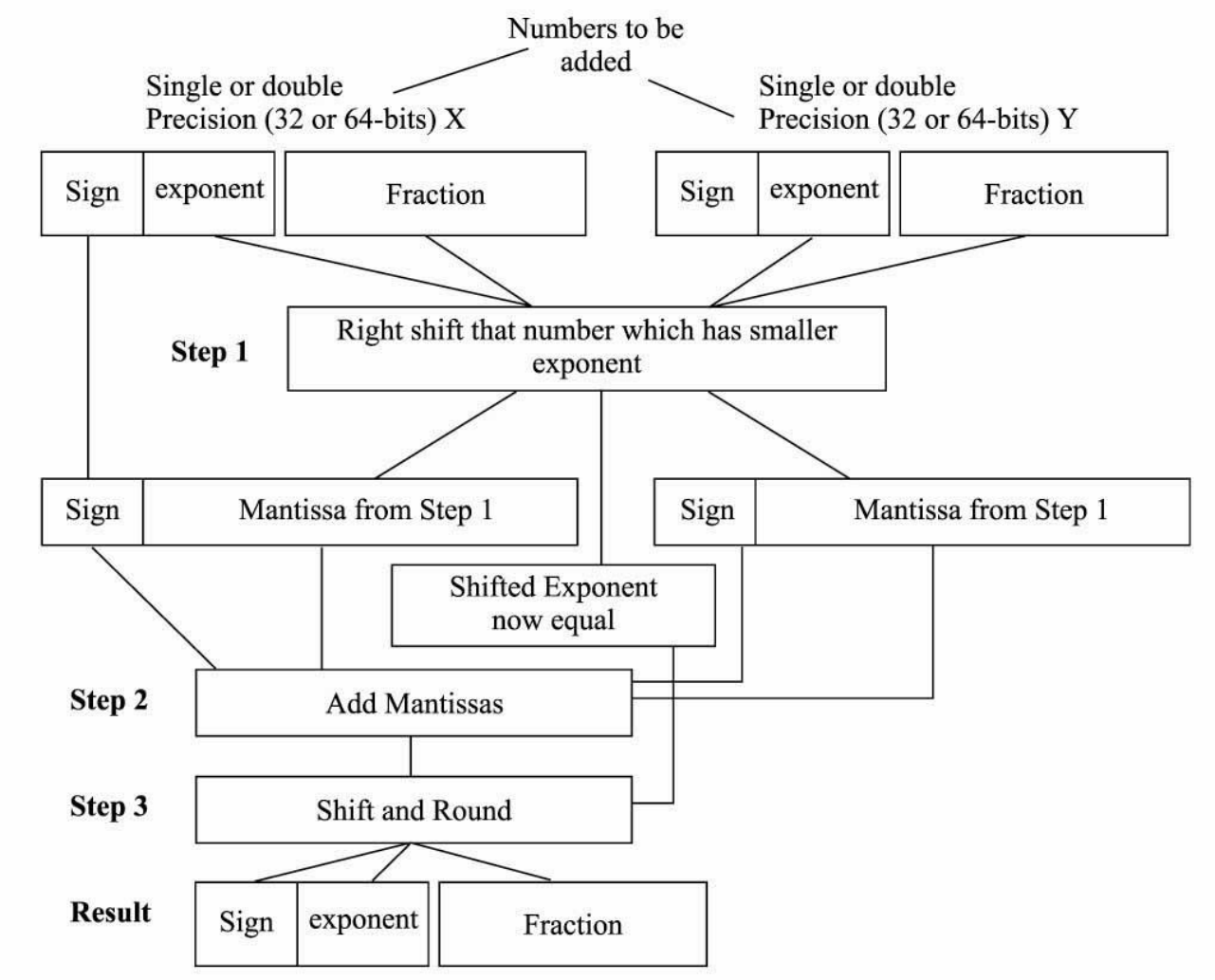
Next steps

- Once the inputs have been shifted, their mantissas are added, and the result shifted if necessary
- Finally, the result is rounded to fit in the fraction field, and the computation is complete

Adding $1.01_2 \times 2^3$ and $1.001_2 \times 2^0$

- Smaller value is shifted to become $0.001001_2 \times 2^3$
- Shifting the number with the smaller exponent allows the use of techniques that retain just enough information about the less-significant bits of the smaller number to perform rounding, reducing the number of bits that actually have to be added

Floating- Point Addition



Subtraction of Floating Point Numbers

Floating-point subtraction

- Uses the same process
- Except that the difference between, rather than the sum of the shifted mantissas is computed

Example of addition of Floating Point Numbers

Compute the sum of 0.25 and 1.5 using single-precision floating-point numbers

- 0.25 = 0b0011 1110 1000 0000 0000 0000 0000 0000 (1.0×2^{-2})
- 1.5 = 0b0011 1111 1100 0000 0000 0000 0000 0000 ($1.5 \times 2^{0)}$
- To add these numbers, shift the one with the smaller exponent (0.25) to the right until both exponents are the same (two places in this case)

Next steps

- This gives mantissas of 1.100 0000 0000 0000 0000 0000 and 0.010 000 0000 0000 0000 0000 for the two numbers (including the assumed 1s in the values to be shifted
- Adding these two mantissas gives a result of 1.110 0000 0000 0 0000 0000 $\times 2^0$ (the exponent of the input with the larger exponent) = 1.75
- The single-precision representation of the full result = 0b0011 1111 1110 0000 0000 0000 0000 0000

Result

- The single-precision representation of the full result = 0b0011 1111 1110 0000 0000 0000 0000 0000

Summary

We learnt

- Multiplication of floating point numbers
- Adding exponents and subtract bias
- Multiply mantissa
- Shift and round
- Division
- Addition by shifting lower number

End of Lesson 09 on Arithmetic using floating point numbers