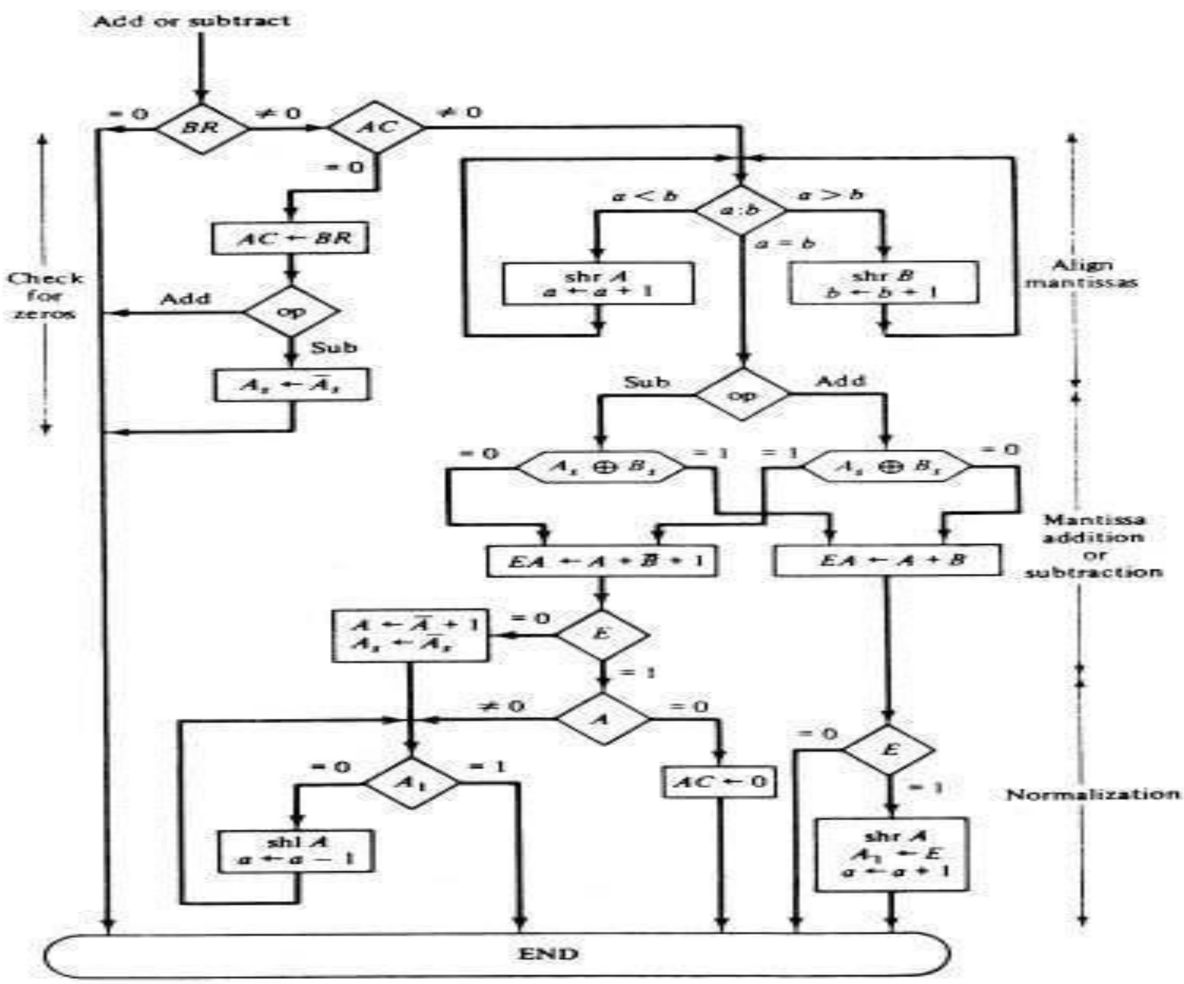
COMPUTER ORGANIZATION AND ARCHITECTURE
UNIT –III
TOPIC- FLOATING POINT ADDITION & SUBTRACTION PART-2

Floating point Addition & Subtraction
Flowchart

- During addition and subtraction, the two floating point operands are in AC and BR. The sum or difference is formed in the AC.

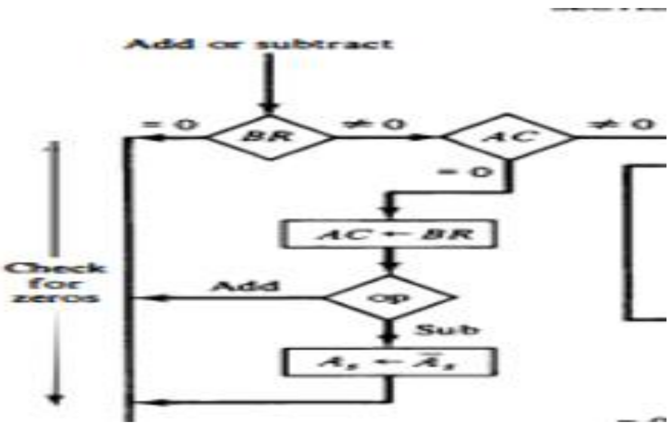
- The algorithm can be divided into four consecutive parts :

1. Check for zeros.
2. Align the mantissa.
3. Add or subtract the mantissa.
4. Normalize the result.



1. Check for zeros.

AC=0	AC=0.00	AC=0.00
BR=0	BR=0.125	BR=-125
AC=0	AC←BR	AC←-BR
		(Reversing sign bit)



2**. Align the mantissa.**

AC=0.538123X10^3
BR=0.123000X10^-1

Align mantissa i.e.,exponent values must be equal.

Difference between 3 & -1 is 4
[3-(-1)]=4

Approach 1: perform **shift left** operation(making first number exponent as 10^-1

AC=0.538123X10^3
AC=0.230000X10^-1
BR=0.123000X10^-1

Data will be lost(MSB)

Decrement exponent by 4

Approach 2: perform **shift right** operation(making 2nd number exponent as 10^3

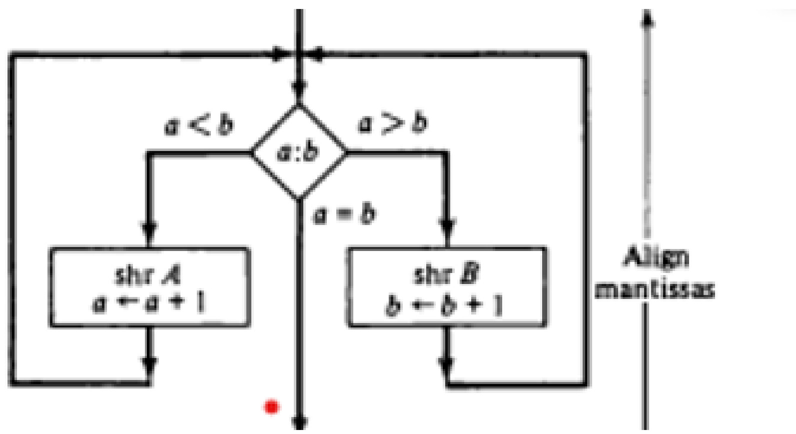
AC=0.538123X10^3

BR=0.123000X10^-1

BR=0.000012X10^3

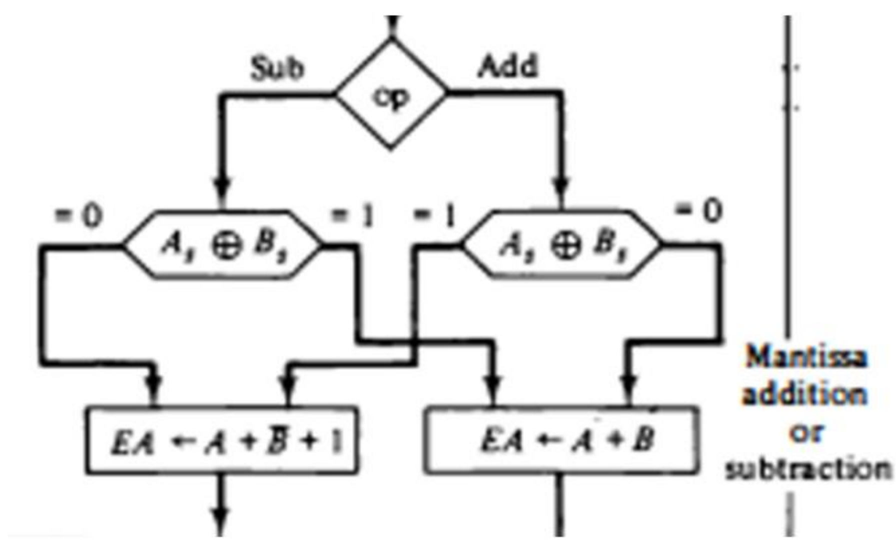
Increment exponent by4

Data will not be lost(LSB)



2. **Add or subtract the mantissa.**

0.532x10^3	0.532x10^3
+ 10.712x10^3	- 0.521x10^3
1.244x10^3	0.011x10^3



4. Normalize the result.

• A floating point number that has a 0 in the most significant position of the mantissa is said to have an UNDERFLOW.

• To normalize a number that contains an underflow, it is necessary to shift the mantissa to the left and decrement the exponent until a nonzero digit appears in the first position

0.532x10^3
+ $_1$0.712x10^3
1.244x10^3

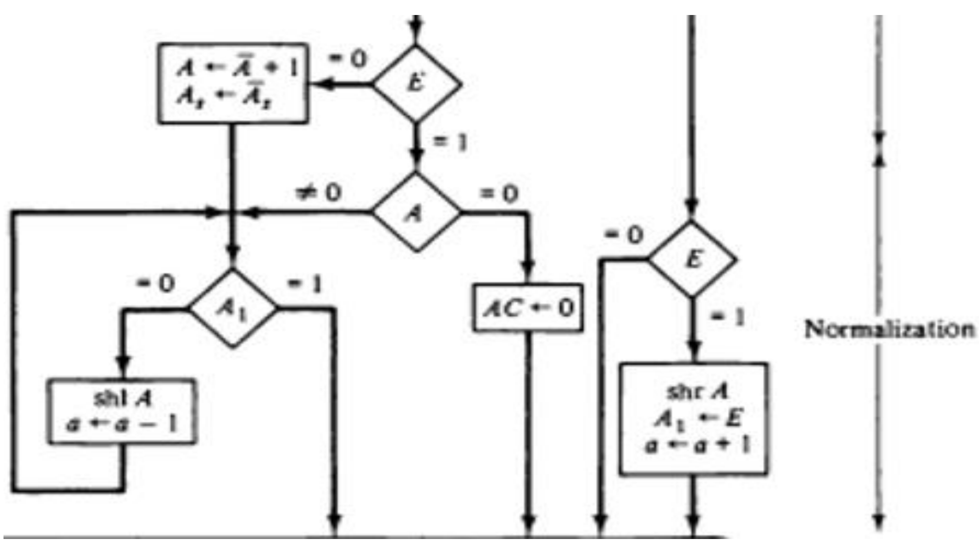
overflow

shift right
1.244x10^3
0.124x10^4(increment exponent)

0.532x10^3
- 0.521x10^3
0.011x10^3

underflow

shift left
0.011x10^3
0.110x10^2(decrement exponent)



• If BR is equal to zero, the operation is terminated, with the value in the AC being the result.

• If AC is equal to zero, we transfer the content of BR into AC and also complement its sign if the numbers are to be subtracted.

• If neither number is equal to zero, we proceed to align the mantissas.

• The magnitude comparator attached to exponents a and b provides three outputs that indicate their relative magnitude. If the two exponents are equal, perform the arithmetic operation.

• If the exponents are not equal, the mantissa having the smaller exponent is shifted to the right and its exponent incremented.

• This process is repeated until the two exponents are equal.

• The addition and subtraction of the two mantissas is identical to the fixed-point addition and subtraction algorithm.

• The magnitude part is added or subtracted depending on the operation and the signs of the two mantissas.

• If an overflow occurs when the magnitudes are added, it is transferred into flip-flop E.

• If E is equal to 1, the bit is transferred into A1 and all other bits of A are shifted right. The exponent must be incremented to maintain the correct number.

• If the magnitudes were subtracted, the result may be zero or may have an underflow.

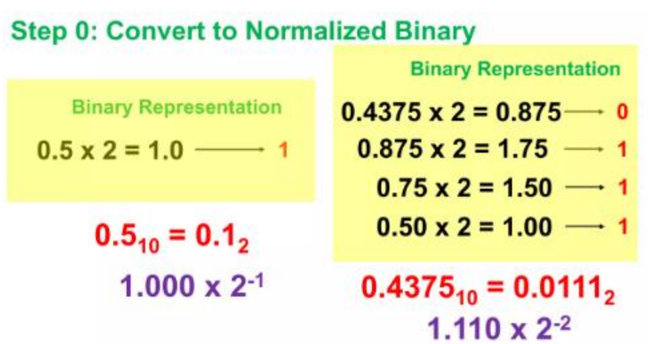
• If the mantissa (A) is zero, the entire floating-point number in the AC is made zero. Otherwise, the mantissa must have at least one bit that is equal to 1.

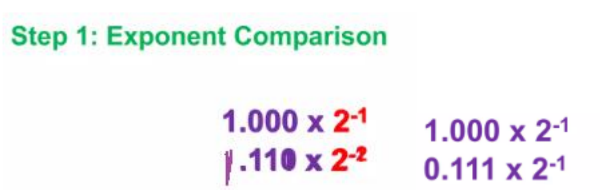
• The mantissa has an underflow if the most significant bit in position A_1 is 0.

• In that case, the mantissa is shifted left and the exponent decremented.

• The bit in A1 is checked again and the process is repeated until it is equal to1. When A1 = 1, the mantissa is normalized and the operation is completed.

Example of floating point addition





Step 2: Addition

$$\begin{array}{r} 1.000 \\ 0.111 \; (+) \\ \hline 1.111 \end{array} \qquad 1.111 \times 2^{-1}$$

Step 3: Normalization

$$1.111 \times 2^{-1}$$

There is an overflow out of the result.

To normalize the result, perform a shift right operation and increment the exponent.

Final Answer

$$1.111 \times 2^{-1} = 0.1111$$

$$0.9375_{10}$$