Chapter 4

Real Number Representations

IEEE754 Floating Point (FP)

A floating-point (FP) representation is used to represent real numbers.

- **The floating-point representation is encoded in a finite number of bits.**

- **The IEEE developed the Floating-point Standard 754 to represent the real numbers.**

- **It was developed in 1985 to standardize computation among the various computer manufactures. The IEEE 754 dictates the precision, accuracy, and arithmetic operations that must be implemented in conforming processors.**

IEEE754 Binary Floating- Point (BFP)

The representation of the IEEE 754 BFP number consists of three parts:

S	EXPONENT	FRACTION

Consider a IEEE 754 BFP number X, it will represented by 3 fields:

i) Sign S_x : is a sign bit and indicates whether the FP number X is positive or negative, (S_x =0 : means X is positive, S_x =1 :means X is negative).

ii) Exponent E_x: Exponent E_x is used to adjust the position of the binary point (as opposed to a "decimal" point. The number of bits of the exponent field (E_x) depends on the format used. The exponent is a signed integer value that can be represented by biased.

The bias B is given by

$$B = 2^{fe\,-1} - 1 \qquad (4.1)$$

Where f_e: is the number of exponent bits in FP format.

Note: Using the biased B is very important to make all exponents E_x in the BFP representation, to be positive number.

iii) Magnitude M_x: IEEE754 BFP standard also calls the Magnitude (M_x) to be a Normalized Significand (or Mantissa)

What is the meaning of Normalized Significand M_x ?

It means that the biased exponent E_x is chosen such that the highest order bit (*Integer bit*) in the significand (M_x) is a 1 (**except for zero value**).

Thus, the normalized significand is represented by

$M_x = 1.F$ with $1 \le M_x \le 2 - 2^{-f}$ or $1 \le M_X < 2$ (4.2) where

F: is the fraction of the real number and it consists of (f- bits). The number of bits of F depends on the format used.

$$F = f_{-1}f_{-2}f_{-3}\ \dots\dots..\ f_{-m}$$

Thus the normalized mantissa is

$$M_x = 1.F = 1.f_{-1}f_{-2}f_{-3}\ \dots\dots..\ f_{-m} \qquad (4.3)$$

Note: The most significant 1 (*integer bit*) is hidden bit (i.e. this integer bit (1) is not stored in IEEE754 BFP registers)

Normalized Representation of IEEE754 BFP Number

S | EXPONENT |	FRACTION
S_X E_X	F

The three fields are packed into one word with the order of fields:

S_x , E_x , and F, such that:

$$X = (-1)^{S_x} . (1.F) . 2^{E_x - B} \qquad \text{Normalized Number} \qquad (4.4)$$

$$\text{let } e_x = E_x - B$$

e_x : the unbiased exponent

E_X : the biased exponent

B : Bias (constant value. It is value depends on the type of standard IEEE754 format to represent BFP number.

Normalized Number

$$X = (-1)^{S_x} . (1.F) . 2^{e_x} \qquad (4.5)$$

IEEE754 Binary Floating Point (BFP) Format

IEEE754 storage format specifies how a **BFP number** is stored in a memory and in the registers of BFP unit.

- **The IEEE 754 BFP standard defines two basics formats:**

a) Single Precision (32- bit)

b) Double Precision (64- bit)

Extended formats for each of these two basics formats are also used. Figure (4.1) shows the data formats supported by the IEEE 754.

S_x	E_x	F

0 1 9 31

(a) Single- Precision.

S_x	E_x	F

0 1 12 43

Fig. (**(b) Extended- Single- Precision.**

S_x	E_x	F
0 1	12	63

(c) Double- precision.

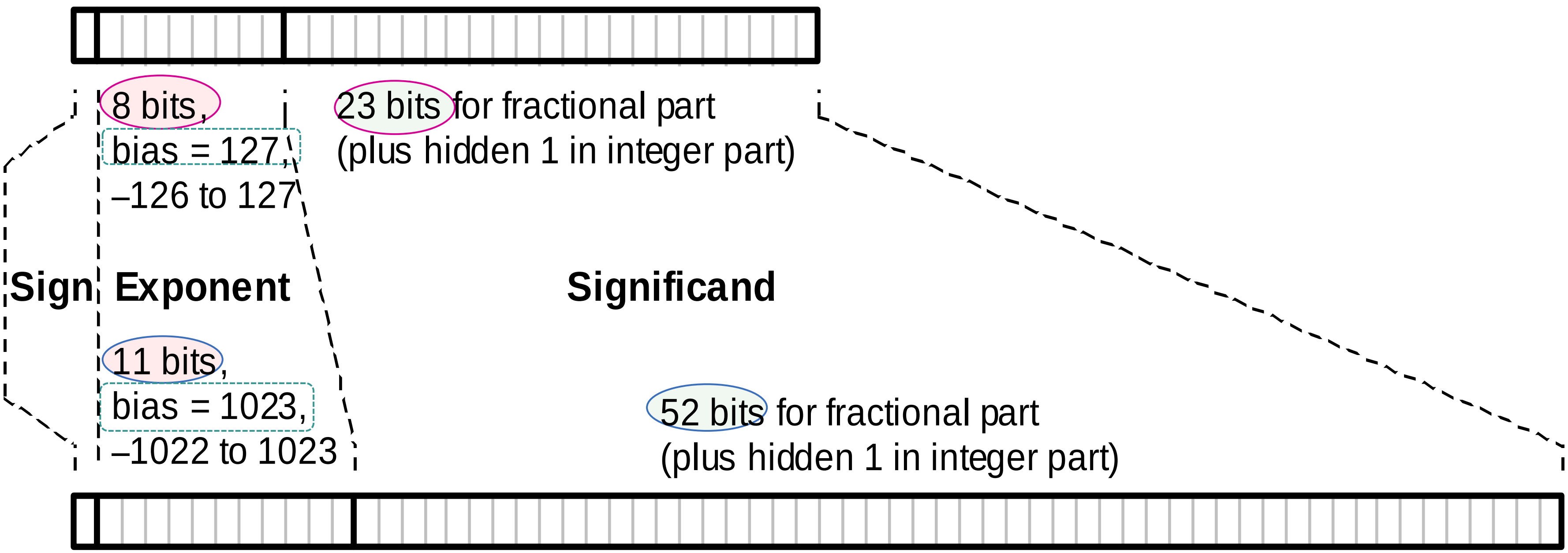
S_x	E_x	F
0 1	16	79

(d) Extended- single- precision.

Fig.(4.1) Data formats supported by the IEEE 754 FP standard representation.

IEEE754 BFP Formats

Single Precision Format: (32- bit)



Double Precision Format: (64-bit)

Special Values

These are the values that are not representable in BFP system, but are useful for representing $\pm\infty$ and **n**ot **a** **n**umber (NaN).

NaN: is a special value, it is useful for representing undefined results, such as (0/0) and the square root of negative number, or when variables are uninitialized.

Example 1: If the biased exponent of X is:

$E_x = 11111111$ (for Single precision format: $E_x(8-bit)$)

$E_x = 11111111111$ (for Double precision format: $E_x(11-bit)$)

Then X is NaN

Special Values

Example 2: Suppose Y is represented in single precision BFP format.

* If all the bits of the biased exponent E_Y are equal 1 and all the fraction bits (F) are equal 0;

then the number Y will be either $-\infty$ or $+\infty$ depending on the sign S_Y:

Biased Exponent is:

$E_Y = 11111111$ $\quad\quad (E_Y: (8 - bit)) \equiv (255)_{10}$

Fraction

$F = 00000000 \ldots 000000$ $\quad\quad (F: (23 - bit)) \equiv (1023)_{10}$

Then Y : $\underline{\pm}\infty$

Features of IEEE754 BFP Floating-Point Formats

Table (4.1)

IEEE 754 FP standard representation and special values

Feature	Single- Precision	Double- Precision
Word width bits	32	64
Significand range	$[1, 2\text{-}2^{-23}]$	$[1, 2\text{-}2^{-52}]$
Exponent bias **(B)**	127	1023
Normalized number **X**	$(-1)^{S_x}.(1.F).2^{E_x-127}$	$(-1)^{S_x}.(1.F).2^{E_x-1023}$
Denormalized Number	$(-1)^{S_x}.(0.F).2^{-126}$	$(-1)^{S_x}.(0.F).2^{-1022}$
Zero Value (± 0)	$(-1)^{S_x}.(1.0)$ and E_x=0	$(-1)^{S_x}.(1.0)$ and E_x=0
NaN	$F \neq 0$ and E_x =255	$F \neq 0$ and E_x =2047
$\pm\infty$ number	$(-1)^{S_x}.(1.0)$ and E_x=255	$(-1)^{S_x}.(1.0)$ and E_x=2047

Exceptions

Five types of exceptions are defined in IEEE 754 BFP Standard. By default, these exceptions set flags and computations continue. The exceptions are:

1- Overflow (exponent): occurs when the result is too large to be represented.

2- Underflow ((exponent): occurs when the nonzero magnitude of the result is too small to be represented.

3- Division by Zero.

4- Inexact: occurs when infinite- precision result different from FP number.

5- Invalid: set when a NaN result is produced.

Conversion Examples

Example 1: **Covert 100_{10} to BFP using IEEE754 single precision format.**

Solution:

Single Precision format means that the data length is 32 bit with the following fields:

(S: 1-bit, E: 8-bit, and F=23- bit "the integer bit is hidden")

Bias B= 127

Step 1: Convert the value of X to binary: $X = 100_{10} \equiv (0110\ 0100)_2$

Step 2: Write X in a normalized BFP representation form , with Mantissa:

$M_x = 1.F = 1.f_{-1}f_{-2}f_{-3} \ \ldots\ldots\ldots \ f_{-m}.$

Since, $X = 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0 \equiv (0\ 1\ 1\ 0\ 0\ 1\ 0\ 0.0) * 2^0$

Right shift X and increase the exponent: $1.100100 \ \ x\ 2^6$

Thus, $X = 1.1001 \ \ x\ 2^6 \equiv 1.F \ \ x\ 2^{ex}$ (normalized)

Step 3: **Write the BFP number (X) in IEEE 754 Single Precision Format**

* Sign $\mathbf{S_X = 0}$ because X is positive

* The unbiased exponent is : $\mathbf{e_X = 6}$

The biased exponent will be : $\mathbf{E_X = e_X + B}$

$E_X = 6+127 = 133_{10} = \mathbf{(1000\ 0101)_2}$

* Extract the fractional part **F** (23-bit) from mantissa M_X

F= 1001000…000000

Pack the three fields to form IEEE754 BFP number X:

S_X	E_X	F
0	**100 0 010 1**	**1 00 1000….0000….00**

4 2 C 8 0 0 0 0 in hexadecimal

X= $(42C8\ 0000)_{16}$ is representation for 100_{16}

Example 2: Covert -175_{10} to BFP using IEEE754 single precision format.

Solution:

$|X| = 175_{10} = 128 + 32 + 8 + 4 + 2 + 1 \equiv (1\,0\,1\,0\,1\,1\,1\,1)_2$

Shift X to right to get normalized value:

$X = 1.0101111 \text{ x } 2^7 \equiv 1.F * 2^{ex} \equiv M_X * 2^{ex}$

- $\mathbf{S_X} = 1$ (Negative number)

- Convert the unbiased exponent $\boldsymbol{e_X}$ to **Biased Exponent** $\mathbf{E_X}$:

$\mathbf{E_X} = \boldsymbol{e_X} + \mathbf{B} = 7 + 127 = 134 = (1000\ 0110)_2$

- Extract the Fraction F from mantissa M_X :

F = 0101111000….000

Thus, IEEE754 BFP Representation of X is

X=	1	100 00110	010 1111 0000 ….0

Or in Hex: X= C32F 0000

Example 3: Convert the IEEE754 BFP number Y= C32F 0000_{16} into its decimal value.

Solution:

Step 1: Extract the three fields from the IEEE754 BFP number Y: (S_y , E_y , F):

1	100 00110	010 1111 0000 0000 0000 0000

* $S_y = 1$ means Y is negative number

* **Biased Exponent E_y** = (1000 0110) ≡ 134_{10}

The unbiased exponent e_Y: $e_Y = E_y - B = 134 – 127$

$e_Y = 7$

* **F** = 010111100…0 ➡ so, the Mantissa M_y = **1.0101111…0**

Step 2: Adjust Mantissa M_y by the exponent (e_Y) (i.e. shift the M_y to left by 7- places and decrease the exponent e_Y).

Thus, Magnitude of **Y = $(1010\ 1111.0)_2$ ≡ - 175**

Exercise: Represent the following real numbers in IEEE754 BFP Single Precision format:

* 0.085
* 0.0
* \- 11.35

Dynamic Range

The goal of using FP representation is to increase the dynamic range, with respect to FX representation. This dynamic range is defined as the ratio between the largest and smallest (nonzero and positive) numbers that can be represented.

Dynamic Range

For a BFX representation using n radix r digits for the magnitude, the dynamic range (*DR_{FX}*) is

$$DR_{FX} = r^{n} - 1$$

For the BFP representation (*DR_{FP}*)

$$DR_{FP} = (r^{f} - 1).r^{(r^{(n-f)} - 1)}$$

Where r: is the radix system (r = 2 for binary system)

n: is the IEEE754 precision type (n=32 for Single precision or n = 64 for Double Precision).

f **:** represent the number of fraction bits. It can be shown that the FP dynamic range is much higher than of FX representation.

It can be shown that the FP dynamic range is much higher than of FX representation.

Homework:

Suppose that the system precision is 32 bit and you have a BFX unit and BFP unit, determine the dynamic range for both units.

Chapter 5

Floating-Point Algorithms and Implementation

1-BFP Adder (Add/ Sub)

Consider two BFP numbers X and Y such that

$$X = (S_X, E_X, M_X) \text{ and } Y = (S_Y, E_Y, M_Y)$$

We consider the basic algorithm for Addition/ (or Subtraction) the two numbers, such that

$Z = X \pm Y$ where $Z = (S_Z, E_Z, M_Z)$ ***Z: Normalized Result***

The Algorithm Steps are:

Step1: **Subtract Exponents:** $d = E_X - E_Y$ (d: called **Alignment shift amount**)

Step2: **Align Significand. This step consists of the following:**

- Shift to right the significand of the operand that has the smallest exponent E by *d*- positions.
- Select as the exponent of the result E_Z such that:

$$E_Z = max(E_X, E_Y)$$

The Algorithm Steps are (continue)

Step3: **Add (/ or subtract) significands (M_x and M_y)** and **produce sign of result S_z**. **This operation is a signed addition.**

The Effective Operation EOP(add or subtract) is determined by the floating-point operation given (ADD or SUBTRACT) and the signs of the operands, as follows:

Floating-Point Operation	Signs of Operands	Effective Operation (EOP)
ADD	equal	add
ADD	different	subtract
SUBTRACT	equal	subtract
SUBTRACT	different	add

$$EOP = S_x \forall S_y \forall op$$

The sign of the result S_Z depends on the signs of the operands (X and Y), the operation, and the relative magnitude of the operands.

Step4: **Normalize the result Z.**

Example1: Add the following two IEEE 754 BFP numbers:

$X = 42C8\ 0000 \equiv 100_{10}$ $\quad\quad$ $Y = 41C8\ 0000 \equiv 25_{10}$

Solution: Extract each number to its three fields:

a) X= 42C8 0000 ≡ 0 100 0010 1 100 1000 0000 …0

Thus, $S_X = 0$ $\quad\quad$ $\boxed{E_x = 1000\ 0101_2} = 133_{10}$

F= 100 1000 0000 …0 ➡ $\boxed{M_x = 1.\ F = 1.10010000\ldots\ 0_2}$

b) Y= 41C8 0000 ≡ 0 100 0001 1 100 1000 0000 ….0

$S_Y = 0$ $\quad\quad$ $\boxed{E_Y = 1000\ 0011_2} = 131_{10}$

F= 100 1000 0000 ….0 ➡ $\boxed{M_Y = 1.\ F = 1.10010000\ldots\ 0_2}$

Add (/Sub) Algorithm Steps

1) **Subtract exponents**: $d = E_X - E_Y$ = (1000 0101_2 - 1000 0011_2) = 10_2

or simply $d = 133_{10} - 131_{10} = 2$

$d = 2$: Alignment Shift Amount

2) Since M_Y is the significand which has the smaller exponent , thus, M_Y needs to be aligned by d - positions (i.e. shift right M_Y by 2 places). Thus, M_Y becomes:

M_Y^* = 0.0110010000… 0_2

3) Add (/subtract) significands , such that $\mathbf{M_Z = M_X\ (EOP)\ M_Y^*}$,

(the EOP in this example is : EOP = ADD)

$$
\begin{array}{lcl}
\mathbf{M_X} & = & \mathbf{1.1001000000\ldots0} \\
\mathbf{M_Y^*} & = & \underline{\mathbf{0.0110010000\ldots0}}
\end{array}
$$

$$\boxed{\mathbf{M_Z = 1.1111010000\ldots0}}$$

$$\mathbf{E_Z} = \boldsymbol{max(E_X, E_Y) = max(133, 131) = 133_{10}}$$

$$\boxed{\mathbf{E_Z \equiv 1000\ 0101_2}}$$

4) Constructing the normalized Result Z:

$S_Z = 0$

$E_Z = 10000101$

M_Z = 1.F = 1.1111010000 ➡ F= 1111010000……..0

Z = $\boxed{0}\boxed{100\ \ 0010\ \ 1}\boxed{111\ \ 1010\ \ 0000\ 0000\ 0000\ 0000}$

Z = $(4\ 2\ F\ A\ 0\ 0\ 0\ 0)_H \equiv 125_{10}$

Example2: Perform the following IEEE 754 BFP operation: $Z = X - Y$

where $X = 4208\ 0000 \equiv 34_{10}$ $Y = 4184\ 0000 \equiv 16.5_{10}$

Solution: Open each number to its three fields:

$X = 4208\ 0000 \equiv 0\ 100\ 0010\ 0\ 000\ 1000\ 0000 \ldots 0$

Thus, $S_X = 0$ $E_x = 1000\ 0100_2$ $= 132_{10}$

$F = 000\ 1000\ 0000 \ldots 0$ $\rightarrow$ $M_x = 1.F = 1.00010000\ldots 0_2$

$Y = 4184\ 0000 \equiv 0\ 100\ 0001\ 1\ 000\ 0100\ 0000 \ldots.0$

$S_Y = 0$ $E_Y = 1000\ 0011_2$ $= 131_{10}$

$F = 000\ 0100\ 0000 \ldots.0$ $\rightarrow$ $M_Y = 1.F = 1.00001000\ldots 0_2$

Add (/Sub) Algorithm Steps

Step 1: **Align the exponents if they are not equal by shifting the smallest number to right by d- positions:**

Thus, subtract exponents: $d = E_X - E_Y = (1000\ 0100_2 - 1000\ 0011_2) = 1_2$

or simply $d = 132_{10} - 131_{10} = 1$

$d = 1$ **: Alignment Shift Amount**

Step2: Right shift M_Y by d- position, because it is exponent is the smaller exponent, thus, (i.e. shift right M_Y by 1 place). Thus, M_Y becomes:

$M_Y^* = 0.100001000\ldots0_2$

Step3: Add (/subtract) significands , such that $\mathbf{M_Z = M_X\ (EOP)\ M_Y^*}$,

(the EOP in this example is : EOP = SUB)

$$
\begin{aligned}
EOP &= S_x \forall S_y \forall Op \\
&= 0 \forall 0 \forall 1 = 1 \\
\therefore\ EOP &\equiv SUB\ operation
\end{aligned}
$$

Thus, take the 1's complement for $\mathbf{M_Y^*}$ to perform subtraction operation:

$\mathbf{M_Y^{*\prime}} = 1.0111101111111111 \dots 1$

Now perform the subtraction operation using the 2's complement addition:

$$
\begin{aligned}
M_X &= 1.0001000000000000 \ \dots.000 \\
M_Y^{*\prime} &= 1.0111101111111111 \dots 11 \\
& \qquad\qquad\qquad\qquad\qquad\qquad 1\ +
\end{aligned}
$$

this is the C_{in}

$M_Z = \boxed{0.10001100000...000000000}$

$E_Z = max(E_X, E_Y) = max(132,131) = 132_{10}$

$E_Z \equiv 1000\ 0100_2$

Note: **The result is not normalized. Thus we should normalize it by shifting the mantissa M_Z to left one position and update the exponent E_Z.**

4) Constructing the normalized Result Z:

$S_Z = 0$

$E_Z = 10000011 \equiv 4_{10}$

M_Z = 1.F = 1.00011000.0000 ⟹ F= 000110000……..0

Z =

S	E	F
0	100 0001 1	000 1100 0000 0000 0000 0000

$Z = (4\ 1\ 8\ C\ 0\ 0\ 0\ 0)_H \equiv 17.5_{10}$

Exercise: **Perform the following IEEE 754 BFP operation: Z= X - Y where X= 4208 0000 $\equiv -0.1875_{10}$ Y= 4184 0000 $\equiv 4.5_{10}$**

2- IEEE754 BFP Multiplication

Consider two BFP numbers X and Y such that

$$X = (S_X, E_X, M_X) \qquad \text{and} \qquad Y = (S_Y, E_Y, M_Y)$$

Multiplication Operation:

$$Z = X * Y \qquad \text{where} \quad Z = (S_Z, E_Z, M_Z)$$ *normalized number*

The Basic Algorithm Steps are:

Step 1: Determine: sign of result, add exponents, and multiply the significands (or Mantissas):

* $S_Z = S_X \forall S_Y$
* $E_Z = E_X + E_Y - B$ **or** $E_Z = e_X + e_Y + B$

  where B: is the bias (B=127 for Single Precision)
* $M_Z = M_X * M_Y$

Step 2: Normalize the significand result M_Z and update the exponent E_Z

Why subtracting the Bias (B) when computing the biased exponent of E_Z?

Ans.

Since the exponents are added in multiplication operation, let us assume that the resultant exponent after multiplication is:

$$E_Z = E_X + E_Y$$
$$= (e_x + B) + (e_y + B)$$
$$= (e_x + e_y) + 2B$$

$\cancel{E_z = e_z + 2B}$ **Incorrect Biased Exponent**

Correct Biased Exponent is

$$\boxed{E_Z = E_X + E_Y - B}$$
$$= (e_x + B) + (e_y + B) - B$$
$$= (e_x + e_y) + 2B - B$$
$$\boxed{E_z = e_z + B}$$

Example1: Multiply the following two FP numbers using IEEE754 FP multiplication:

$X \equiv (0.29 * 10^2)_{10}$ $Y = (1.12 * 10^2)_{10}$

Solution:

Aside:

$$Z \equiv (X * Y) = (0.29 * 10^2)_{10} * (1.12 * 10^2)_{10} = 0.3248 * 10^4$$

Convert X and Y to BFP numbers:

$X = (0.29 * 10^2) = 29_{10} \equiv (11101)_2 = 1.1101 * 2^4 \equiv 1.F \,.\, 2^{e_x}$

$Y = (1.12 * 10^2) = 112_{10} \equiv (1110000)_2 = 1.110000 * 2^6 \equiv 1.F \,.\, 2^{e_y}$

Return

Multiplication Algorithm:

Step 1:

***Sign of Result:** $S_Z = S_X \forall S_Y = 0 \forall 0 = 0$

- **Exponent of Product:** $E_Z = (E_X + E_Y - B)$ Or $(\boldsymbol{e_x} + \boldsymbol{e_y} + \mathbf{B})$

$$= 4 + 6 + 127 = 137_{10}$$

$$\therefore E_Z \equiv 137_{10} \equiv (10001001)_2$$

- **Mantissa:** $M_Z = M_X * M_Y = (1.1101 * 1.110000)$

$$M_Z = 11.0010110000$$

$\boldsymbol{M_Z \geq 2}$ *Thus, its not normalized*

: It is overflow result

Step 2: Normalize the *Mantissa* M_Z by shifting it one position to right and increment the exponent E_Z

$$M_Z = 1.10010110000$$

$$E_Z \equiv 137_{10} + 1 = 138_{10} \equiv (10001010)_2$$

$$\equiv (0.3248 * 10^4)_{10}$$

Z =

0	**10001010**	**10010110000 … … 00**

Example2: Multiply the following two FP numbers using IEEE754 FP multiplication:

$X \equiv 43498000 = 201.5_{10}$ $\quad$ $Y = C1600000 = -14_{10}$

Solution: $Z \equiv X * Y = (201.5 * (-14))_{10} = -2821.0$

Aside:

Extract the three fields of each operand:

$X \equiv 0\ 10000110\ 10010011000 \ldots 0000$

$\therefore\ S_x = 0,\ \ E_x = 134_{10} \quad or \quad e_x = 7_{10}\ (unbiased\ exponent),$
$and\ \ M_x = 1.F = 1.1001001100000 \ldots 000$

$Y \equiv 1\ 10000010\ 1100000000..000000$

$\therefore\ S_y = 1,\ \ E_y = 130_{10} \quad or \quad e_y = 3_{10}(unbiased\ exponent),$
$and\ \ M_y = 1.F = 1.1100000000000 \ldots 000$

Return

3- IEEE754 BFP Division

Consider two BFP numbers X and Y such that

$$X = (S_X, E_X, M_X) \qquad \text{and } Y = (S_Y, E_Y, M_Y)$$

Division Result is :

$$Z = X/Y \qquad \text{where} \quad Z = (S_Z, E_Z, M_Z) \text{ normalized number}$$

The Basic Algorithm Steps are:

Step1: Check if either one or both operands are equal to zero. If Y=0, a Division by zero **flag is set**. If no, perform step 2

Step2: Determine The sign of result, subtract exponents, and divide significands:

$$S_Z = S_X \forall S_Y$$

$$E_Z = E_X - E_Y + B \quad \text{or} \quad E_Z = e_X - e_Y + B$$

$$M_Z = M_X / M_Y$$

Step3: Normalize M_Z and update the exponent E_Z if necessary

Example1: Divide the following two IEEE 754 BFP numbers:

$X \equiv (C464000)_{16}$ $\qquad\qquad$ $Y = (45640000)_{16}$

Solution: $Z = \left(\frac{X}{Y}\right)$

Extract each number to its three fields:

$X \equiv (C464000)_{16}$

$= \boxed{1}\ \boxed{10001000}\ \boxed{110010000....000}$ $\Rightarrow$ $E_X = 10001000_2 = 136_{10}$

$Y = (45640000)_{16}$

$= \boxed{0}\ \boxed{10001010}\ \boxed{110010000....000}$ $\Rightarrow$ $E_Y = 10001010_2 = 138_{10}$

Step1: Check if M_x or M_y or both equal to zero

in this example: M_x and $M_y \neq 0$

Step2: **Determine The sign of result, subtract exponents, and divide significands:**

* $S_Z = S_X \forall S_Y = 1 \forall 0 = 1$ **Result is negative**

* $E_Z = E_X - E_Y + B$ **or** $E_Z = e_X - e_Y + B$

$E_Z = 136 - 138 + 127 = 125_{10} \equiv (01111101)_2$

* $M_Z = M_X / M_Y$

$= 1.110010000....000 / 1.110010000....000$

$M_Z = 1.000000000....000$

Step3: **Normalize M_Z (if needed) and update the exponent E_Z if necessary:**

M_Z is already normalized. Thus the result of division Z is:

	S_Z	E_Z	F
Z =	1	01111101	000000000…0000

Example2: Divide the following two IEEE 754 BFP numbers:

$X \equiv (42B6B000)_{16} \equiv 91.34375_{10}$, $Y = (3E140000)_{16} \equiv 0.14453125_{10}$

Solution: $Z = \left(\frac{X}{Y}\right)$

Extract each number to its three fields:

$X \equiv (42B6B000)_{16}$

=

0	**10000101**	**011011010110..00**

➡ $E_X = 10001000_2 = 133_{10}$

or $e_x = E_x - B = 133 - 127 = 6_{10}$

$Y = (3E140000)_{16}$

=

0	**01111100**	**00101000000..000**

➡ $E_Y = 10001010_2 = 124_{10}$

or $e_y = E_y - B = 124 - 127 = -3_{10}$

Step1: Check if M_x or M_y or both equal to zero

in this example: M_x and $M_y \neq 0$

If the result is negative, convert the mantissa back to signed magnitude by inverting the bits and adding 1.

Solution: $\mathbf{Z} = \left(\frac{X}{Y}\right)$