Floating Point Range

Special Values (single-precision)

E’	F	Meaning	Notes
00000000	**0…0**	**0**	**+0.0 and -0.0**
00000000	**X…X**	**Valid number**	**Non normalized or denormal number = (-1)Sx 2^(-126)x (0.F)**
11111111	**0…0**	**Infinity**	
11111111	**X…X**	**Not a Number**	

Range of numbers

- Normalized (positive range; negative is symmetric)

smallest	0	00000001	00000000000000000000000	$+2^{-126}(1+0) = 2^{-126}$
largest	0	11111110	11111111111111111111111	$+2^{127}(2-2^{-23})$

- Unnormalized

smallest	0	00000000	00000000000000000000001	$+2^{-126}(2^{-23}) = 2^{-149}$
largest	0	00000000	11111111111111111111111	$+2^{-126}(1-2^{-23})$

Compare FP numbers (<, > ?)

Examples:

1 . A= 0 0111 1111 110…0 B=0 1000 0000 110…0

$+(1.11)_2 \times 2$^ (127-127) =1.750 $+(1.11)_2 \times 2$ ^(128-127)= $(11.1)_2$=3.500

0 0111 1111 110…0 0 1000 0000 110…0

+0111 1111 < + 1000 0000 implies B>A

directly comparing exponents as unsigned values gives result

2. A= 1 0111 1111 110…0 B= 1 1000 0000 110…0

-f ×2^(0111 1111) -f ×2^(1000 0000)

For exponents: 0111 1111 < 1000 0000

So -f ×2(0111 1111) > -f ×2(1000 0000)

If (both S=1) and (E'_B > E'_A) then A > B

Floating Point addition is not Associative

$(x + y) + z \neq x + (y + z)$

(1.101100...00x 2^127 - 1.101100..00x 2^127)+1 = 1

1.101100..00x 2^127+ (1 – 1 .101100..00x2^127)=

(1 –1 .101100..00x2^127)=

0.(26zeros 1x 2^127) - 1 .101100..00x2^127

0.(26zeros 1x 2^127)= 0 in single precision fp

1.101100..00x 2^127 - 1.101100..00x 2^127= 0

Assignment

1. $(A\ B\ C\ D\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)_{16}$ is in IEEE double precision floating point format.

Convert to its decimal value.

2. Represent binary +tive number 1101011 in IEEE single precision floating point format.
3. Represent decimal number -0.75 in IEEE single precision floating point format.

Example

Represent $(+7)_{10}$ in IEEE double precision floating point :

1. Convert to binary $(+7)_{10} = (111)_2$
2. Normalized $(111)_2 = 1.11 \times 2^2$
3. 52 bit mantissa = $1\underbrace{10\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots000}_{\text{50 zeros}}$

   50 zeros

4. Biased exponent in excess 1023 = 2+1023=1025

   2^{10} 2^{0}

$0\ \underbrace{1\ 000\ 000\ 0001}_{\text{Biased exponent in excess 1023}}\ 1\underbrace{10\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots000}_{\text{50 zeros}}$

Biased exponent in excess 1023

50 zeros

Floating Point Rep --Underflow

- Result of arithmetic operation on floating point number too small to be stored in computer then underflow
- 2 floating point numbers are subtracted

If at least one zero is in most significant position of mantissa then underflow

i.e. result= 0.0001111 x 2 is underflow

Corrected to 1.111×2^{-4}

By left shift and decreasing exponent until non zero bit in left most position

Floating Point Rep -- Underflow

- In division of floating point numbers exponents are subtracted

If exponent E’ < 1 or E <-126 (in single precision)and E< - 1022(in double precision) then underflow

Cannot be Corrected

Floating Point Rep --Overflow

- Result of arithmetic operation on floating point number too large to be stored in computer then overflow
- 2 floating point numbers of same sign are added.

If carry from most significant position

then mantissa overflow

i.e. result= 10.1111 x 2 is overflow

Corrected to 1.01111×2^2

By right shift and increasing exponent by same amount

Floating Point Rep -- Overflow

- In multiplication of floating point numbers exponents are added

If exponent E’ > 254 or E > 127 (in single precision)and E’ > 2046 or E> 1023(in double precision) then overflow

Cannot be Corrected

Floating Point Rep of Num

Single precision floating point normalized number with exponent: range

-126<=E<=127, 1<=E’<=254

Double precision floating point normalized number with exponent range

-1022<=E<=1023, 1<=E’<=2046

Special Values :

E’=0,M=0 exact zero value represented

E’=255,M=0 infinity(i.e. divide by zero a normal num)

Two more

Special Values defined for IEEE Standard Floating Point Format (754)

E’=0,M=0 exact zero value represented

E’=255,M=0 infinity(i.e. a normal num divide by zero)

E’= 0, M not =0 denormal numbers

(+/-)0.M x 2^{-126}

Gradual underflow accommodated to handle very

small numbers

E’=255, M not = 0   Not a Number(NaN) – result of zero divided by zero, √-1

How are special values set?

- Processor sets exception flag for:
  - Underflow
  - Overflow
  - Divide by zero (E'=255 set, M=0 set:- infinity)
  - Invalid(if 0/0 or √-1 operation attempted)
    - (E'=255 and M= non zero set :- NaN not a number)
  - Inexact (if rounding off required)
- When exceptions occur, results are set to special value

2's-Complement Overflow

(5 bit **signed integer**: range(-2^{5-1} to $+ 2^{5-1}-1$)

- If *X*, *Y* have opposite signs overflow never occurs whether carry-out exists or not

No Carry-out

$$\begin{array}{lr} 00101 & (+\ 5_{10}) \\ 10110 & (-\ 10_{10}) \\ \hline 11011 & (-\ 5_{10}) \end{array}$$

Carry-out

$$\begin{array}{lr} 01010 & (+\ 10_{10}) \\ 11011 & (-\ 5_{10}) \\ \hline 1\ 00101 & (+\ 5_{10}) \end{array}$$

If *X*, *Y* have same sign and result sign differs, overflow occurs

$$\begin{array}{lr} 11001 & (-\ 7_{10}) \\ 10110 & (-\ 10_{10}) \\ \hline 1\ 01111 & (+\ 15_{10}) \end{array}$$

Carry-out, Overflow

$$\begin{array}{lr} 00111 & (+\ 7_{10}) \\ 01010 & (+\ 10_{10}) \\ \hline 10001 & (-\ 15_{10}) \end{array}$$

No Carry-out, Overflow

Overflow: An Error

- Examples: Addition of <u>3-bit integers </u>(range - 4 to +3)

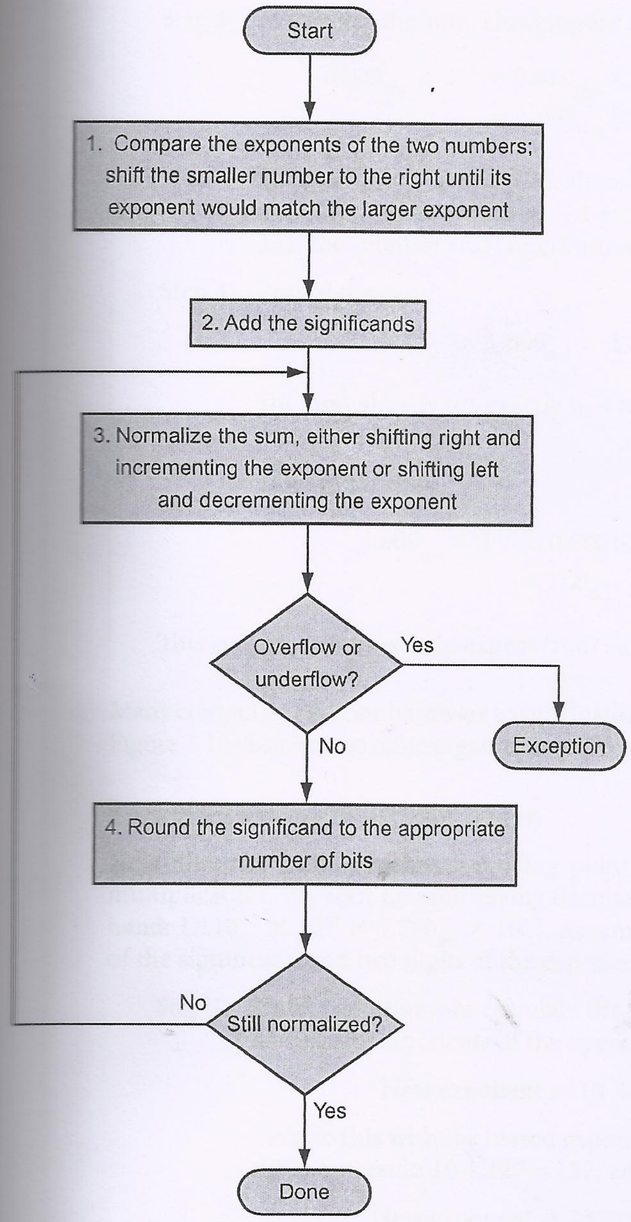
  - -2-3 = -5 110 = -2
    + 101 = -3
    = 1011 = 3 (error)

  - 3+2 = 5 011 = 3
    010 = 2
    = 101 = -3 (error)

- **Overflow rule:**
  - **If two numbers with the same sign bit (both positive or both negative) are added, the overflow occurs if and only if the result has the opposite sign.**
  - **OR Carry-in into MSB ≠ Carryout from MSB**

Floating Pointer Arithmetic Operations

Flow Chart Floating Point Number Addition



Floating-point addition. The normal path is to execute steps 3 and 4 once, but if rounding causes the sum to be unnormalized, we must repeat step 3.

Add/Subtract Rule

1. Choose the number with the smaller exponent and shift its mantissa right a number of steps equal to the difference in exponents.
2. Set the exponent of the result equal to the larger exponent.
3. Perform addition/subtraction on the mantissas and determine the sign of the result.
4. Normalize the resulting value, if necessary.

Flowchart Floatig Point Number Multipli- -cation Patterson et. al.

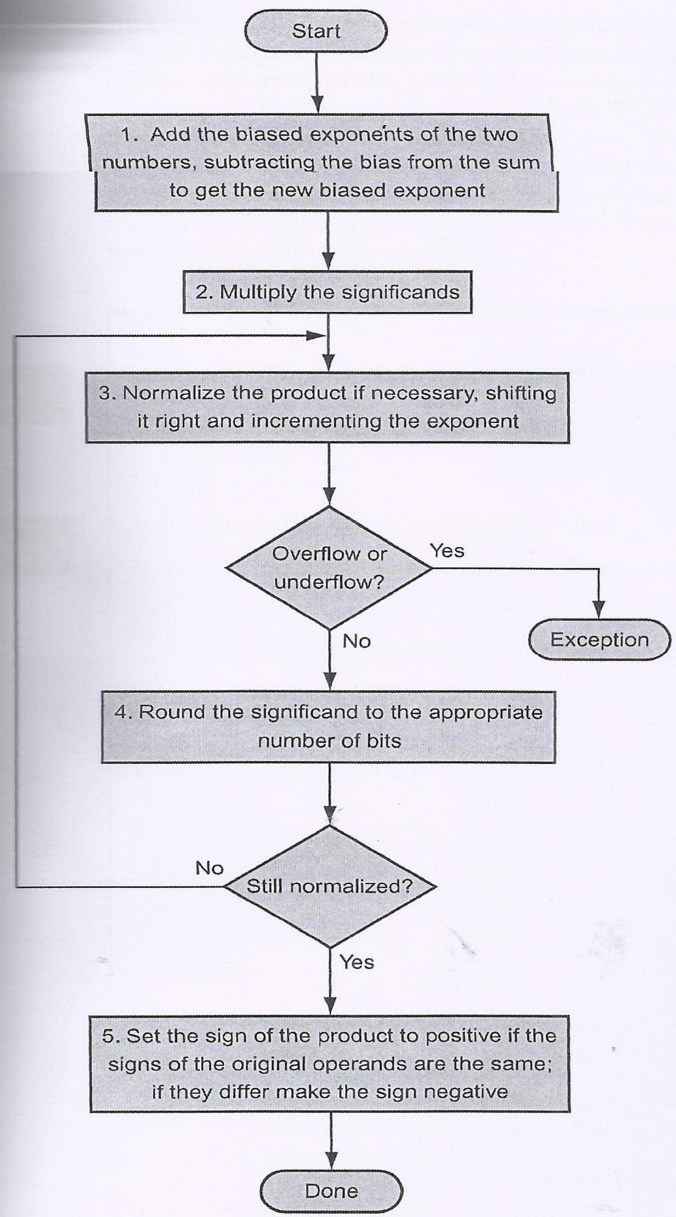


FIGURE 3.16 **Floating-point multiplication.** The normal path is to execute steps 3 and 4 once, but if rounding causes the sum to be unnormalized, we must repeat step 3.

Multiply Rule

1. Add the exponents and subtract 127.
2. Multiply the mantissas and determine the sign of the result.
3. Normalize the resulting value, if necessary.

Divide Rule

1. Subtract the exponents and add 127.
2. Divide the mantissas and determine the sign of the result.
3. Normalize the resulting value, if necessary.

The addition or subtraction of 127 in the multiply and divide rules results from using the excess-127 notation for exponents.

Floating Point Add Subtract Signs

SA	SB	Add/Sub		SR
1	0	1		1
0	0	1	If E’A>E’B	0
0	0	1	If E’A<E’B	1
0	0	1	If E’A=E’B If FA>FB If FA<FB	 0 1
1	1	1		0
1	0	0	If E’A>E’B	1
1	0	0	If E’A<E’B	0
1	0	0	If E’A=E’B If FA>FB If FA<FB	 1 0

Floating Point Add Subtract Signs

0	1	0	If E’A>E’B	0
0	1	0	If E’A<E’B	1
0	1	0	If E’A=E’B If FA>FB If FA<FB	 0 1
0	0	0		0
1	1	0		1