Floating Point Number Representation

Dr. PRADUMN KUMAR PANDEY

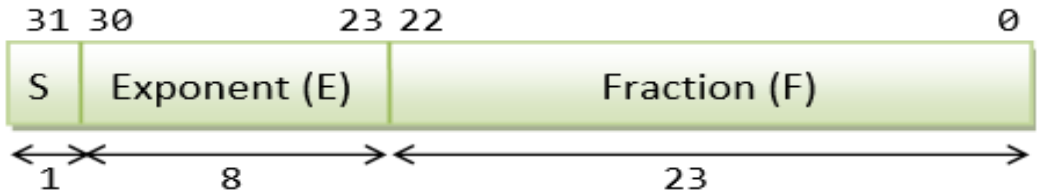
2022-11-11

Floating Point Number Representation

- In computers, floating-point numbers are represented in scientific notation of fraction (F) and exponent (E) with a radix of 2, in the form of $F \times 2^E$.
- Both E and F can be positive as well as negative.
- Modern computers adopt IEEE 754 standard for representing floating-point numbers.
- There are two representation schemes: 32-bit single-precision and 64-bit double-precision.

IEEE-754 32-bit Single-Precision Floating-Point Numbers

- In 32-bit single-precision floating-point representation:
  - The most significant bit is the sign bit (S), with 0 for positive numbers and 1 for negative numbers.
  - The following 8 bits represent exponent (E).
  - The remaining 23 bits represents fraction (F).



32-bit Single-Precision Floating-point Number

Normalized Form

- Let's illustrate with an example, suppose that the 32-bit pattern is 1 1000 0001 011 0000 0000 0000 0000 0000, with:
  - S = 1
  - E = 1000 0001
  - F = 011 0000 0000 0000 0000 0000
- In the normalized form, the actual fraction is normalized with an implicit leading 1 in the form of 1.F.
- In this example, the actual fraction is $1.011\ 0000\ 0000\ 0000\ 0000\ 0000 = 1 + 1\times2^{-2} + 1\times2^{-3} = 1.375\text{D}$.

contd..

- The sign bit represents the sign of the number, with S=0 for positive and S=1 for negative number. In this example with S=1, this is a negative number, i.e., -1.375D.
- In normalized form, the actual exponent is E-127 (so-called excess-127 or bias-127). This is because we need to represent both positive and negative exponent. With an 8-bit E, ranging from 0 to 255, the excess-127 scheme could provide actual exponent of -127 to 128. In this example, E-127=129-127=2D.
- Hence, the number represented is $-1.375 \times 2^2 = -5.5$D.

De-Normalized Form

- Normalized form has a serious problem, with an implicit leading 1 for the fraction, it cannot represent the number zero!
- De-normalized form was devised to represent zero and other numbers.
- For E=0, the numbers are in the de-normalized form. An implicit leading 0 (instead of 1) is used for the fraction; and the actual exponent is always -126. Hence, the number zero can be represented with E=0 and F=0 (because $0.0 \times 2^{-126}=0$).

contd..

- We can also represent very small positive and negative numbers in de-normalized form with E=0.
- For example, if S=1, E=0, and F=011 0000 0000 0000 0000 0000. The actual fraction is $0.011 = 1 \times 2^{-2} + 1 \times 2^{-3} = 0.375$D.
- Since S=1, it is a negative number. With E=0, the actual exponent is -126. Hence the number is $-0.375 \times 2^{-126} = -4.4 \times 10^{-39}$, which is an extremely small negative number (close to zero).

Summary

- In summary, the value (N) is calculated as follows:
  - For $1 \leq \text{E} \leq 254$, $\text{N} = (\text{-}1)^S \times 1.\text{F} \times 2^{(E-127)}$. These numbers are in the so-called normalized form. The sign-bit represents the sign of the number. Fractional part (1.F) are normalized with an implicit leading 1. The exponent is bias (or in excess) of 127, so as to represent both positive and negative exponent. The range of exponent is -126 to +127.

contd..

- For E = 0, N = (-1)^S × 0.F × $2^{(-126)}$. These numbers are in the so-called denormalized form. The exponent of 2^{-126} evaluates to a very small number. Denormalized form is needed to represent zero (with F=0 and E=0). It can also represents very small positive and negative number close to zero.
- For E = 255, it represents special values, such as ±INF (positive and negative infinity) and NaN (not a number).

Example 1

Suppose that IEEE-754 32-bit floating-point representation pattern is 0 10000000 110 0000 0000 0000 0000 0000.

- Sign bit S = 0 $\Rightarrow$ positive number
- E = 1000 0000B = 128D (in normalized form)
- Fraction is 1.11B (with an implicit leading 1)
  $= 1 + 1\times 2^{-1} + 1 \times 2^{-2} = 1.75$D
- The number is $+1.75 \times 2^{(128-127)} = +3.5$D

Example 2

Suppose that IEEE-754 32-bit floating-point representation pattern is 1 01111110 100 0000 0000 0000 0000 0000.

- Sign bit S = 1 $\Rightarrow$ negative number
- E = 0111 1110B = 126D (in normalized form)
- Fraction is 1.1B (with an implicit leading 1) = $1 + 2^{-1} = 1.5$D
- The number is -1.5 $\times$ $2^{(126-127)}$ = -0.75D

Example 3

Suppose that IEEE-754 32-bit floating-point representation pattern is 1 01111110 000 0000 0000 0000 0000 0001.

- Sign bit S = 1 $\Rightarrow$ negative number
- E = 0111 1110B = 126D (in normalized form)
- Fraction is 1.000 0000 0000 0000 0000 0001B (with an implicit leading 1) = $1 + 2^{-23}$
- The number is -$(1 + 2^{-23}) \times 2^{(126-127)}$
  = -0.50000005960464477539062 (??)

Example 4

(De-Normalized Form): Suppose that IEEE-754 32-bit floating-point representation pattern is 1 00000000 000 0000 0000 0000 0000 0001.

- Sign bit S = 1 negative number
- E = 0 (in de-normalized form)
- Fraction is 0.000 0000 0000 0000 0000 0001B (with an implicit leading 0) = 1×2^{-23}
- The number is -$2^{-23} \times 2^{(-126)} =$ -$2^{(-149)} \approx$ -1.4$\times 10^{-45}$