IEEE 754 Floating-Point Format

Floating-Point Decimal Number

$$
\begin{aligned}
-123456. \times 10^{-1} &= 12345.6 \times 10^{0} \\
&= 1234.56 \times 10^{1} \\
&= 123.456 \times 10^{2} \\
&= 12.3456 \times 10^{3} \\
&= 1.23456 \times 10^{4} (\textit{normalised}) \\
&\approx 0.12345 \times 10^{5} \\
&\approx 0.01234 \times 10^{6}
\end{aligned}
$$

Note

- There are different representations for the same number and there is no fixed position for the decimal point.
- Given a fixed number of digits, there may be a loss of precession.
- Three pieces of information represents a number: sign of the number, the significant value and the signed exponent of 10.

Note

Given a fixed number of digits, the floating-point representation covers a wider range of values compared to a fixed-point representation.

Example

The range of a fixed-point decimal system with six digits, of which two are after the decimal point, is 0.00 to 9999.99.

The range of a floating-point representation of the form $m.mmm \times 10^{ee}$ is $0.0, 0.001 \times 10^0$ to 9.999×10^{99}. Note that the radix-10 is implicit.

In a C Program

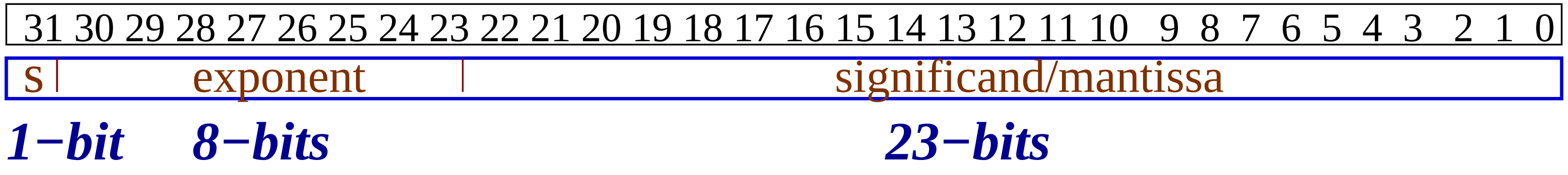
- Data of type `float` and `double` are represented as binary floating-point numbers.
- These are approximations of real numbers[a] like an `int`, an approximation of integers.

[a] In general a real number may have infinite information content. It cannot be stored in the computer memory and cannot be processed by the CPU.

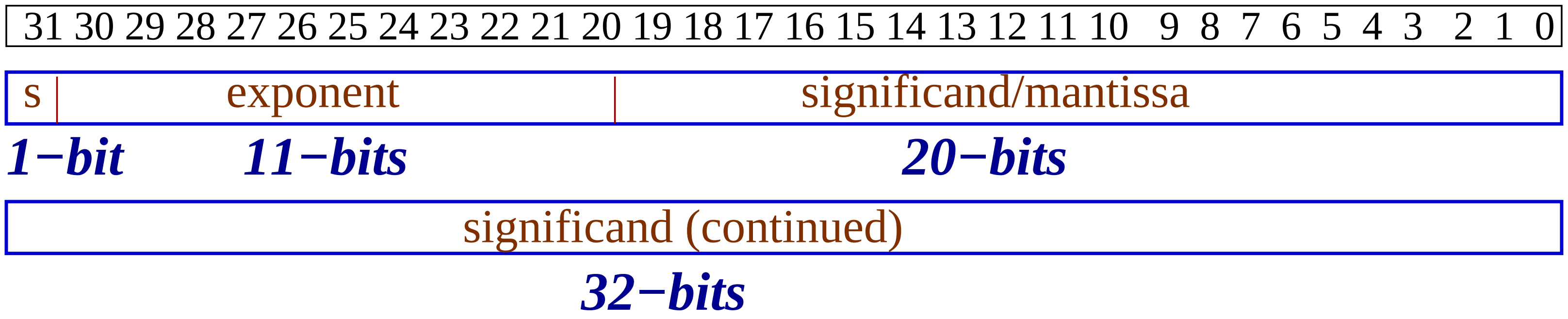
IEEE 754 Standard

Most of the binary floating-point representations follow the IEEE-754 standard. The data type `float` uses IEEE 32-bit single precision format and the data type `double` uses IEEE 64-bit double precision format.

A floating-point constant is treated as a double precision number by GCC.



Single Precession (32−bit)



Double Precession (64−bit)

Single Precession Normalized Number

Let the sign bit (31) be s, the exponent (30-23) be e and the mantissa (significand or fraction) (22-0) be m. The valid range of the exponents is 1 to 254 (if e is treated as an unsigned number).

- The actual exponent is biased by 127 to get e i.e. the actual value of the exponent is $e - 127$. This gives the range: $2^{1-127} = 2^{-126}$ to $2^{254-127} = 2^{127}$.

Single Precession Normalized Number

- The normalized significand is $1.m$ (binary dot). The binary point is before bit-22 and the 1 (one) is not present explicitly.
- The sign bit $s = 1$ for a $-$ve number is zero (0) for a +ve number.
- The value of a normalized number is

$$(-1)^s \times 1.m \times 2^{e-127}$$

An Example

Consider the following 32-bit pattern

1 1011 0110 011 0000 0000 0000 0000 0000

The value is

$$
\begin{aligned}
& (-1)^{1} \times 2^{10110110-01111111} \times 1.011 \\
= \; & -1.375 \times 2^{55} \\
= \; & -49539595901075456.0 \\
= \; & -4.9539595901075456 \times 10^{16}
\end{aligned}
$$

An Example

Consider the decimal number: +105.625. The equivalent binary representation is

$$
\begin{aligned}
& +1101001.101 \\
= \; & +1.101001101 \times 2^{6} \\
= \; & +1.101001101 \times 2^{133-127} \\
= \; & +1.101001101 \times 2^{10000101-01111111}
\end{aligned}
$$

In IEEE 754 format:

0 1000 0101 101 0011 0100 0000 0000 0000

An Example

Consider the decimal number: $+2.7$. The equivalent binary representation is

$$
\begin{aligned}
& +10.10\ 1100\ 1100\ 1100 \cdots \\
= & +1.010\ 1100\ 1100 \cdots \times 2^{1} \\
= & +1.010\ 1100\ 1100 \cdots \times 2^{128-127} \\
= & +1.010\ 1100 \cdots \times 2^{10000000-01111111}
\end{aligned}
$$

In IEEE 754 format (approximate):

0 1000 0000 010 1100 1100 1100 1100 1101

Range of Significand

The range of significand for a 32-bit number is 1.0 to $(2.0 - 2^{-23})$.

Note

- The smallest magnitude of a normalized number in single precession is $\pm$ 0000 0001 000 0000 0000 0000 0000 0000, whose value is 1.0×2^{-126}.
- The largest magnitude of a normalized number in single precession is $\pm$ 1111 1110 111 1111 1111 1111 1111 1111, whose value is $1.99999988 \times 2^{127} \approx 3.403 \times 10^{38}$.

Note

- The smallest magnitude of a subnormal number in single precession is $\pm$ 0000 0000 000 0000 0000 0000 0000 0001, whose value is $2^{-126+(-23)} = 2^{-149}$.
- The largest magnitude of a subnormal number in single precession is $\pm$ 0000 0000 111 1111 1111 1111 1111 1111, whose value is $0.99999988 \times 2^{-126}$.

Note

Infinity:

∞: 1111 1111 000 0000 0000 0000 0000 0000

is greater than (as an unsigned integer) the largest normal number:

1111 1110 111 1111 1111 1111 1111 1111

```
#include <stdio.h>
int main() // twoZeros.c
{
    double a = 0.0, b = -0.0 ;

    printf("a: %f, b: %f\n", a, b) ;
    if(a == b) printf("Equal\n");
    else printf("Unequal\n");
    return 0;
}
```

```
$ cc -Wall twoZeros.c
$ a.out
a:  0.000000, b:  -0.000000
Equal
```

Largest $+1 = \infty$

The 32-bit pattern for infinity is

0 1111 1111 000 0000 0000 0000 0000 0000

The largest 32-bit normalized number is

0 1111 1110 111 1111 1111 1111 1111 1111

If we treat the largest normalized number as an `int` data and add one to it, we get ∞.