Floating Point Number System

A Reference Guide to IEEE 754 Floating Point Formats

1. Introduction

Floating point representation is a method used by computers to encode real numbers within the constraints of finite precision. The IEEE 754 standard, established in 1985 and revised in 2008 and 2019, defines the most widely adopted formats for floating point arithmetic.

A floating point number consists of three components: a **sign bit** (determining positive or negative), an **exponent** (determining the magnitude), and a **mantissa** (also called significand, determining the precision).

General Formula

The value of a normalized floating point number is calculated as:

```
Value = (-1)^S × (1 + M) × 2^(E - bias)
```

Where S is the sign bit, M is the fractional mantissa, E is the stored exponent, and bias is a format-specific constant.

2. Format Comparison

The following table summarizes the three common IEEE 754 floating point formats:

Property	Half (FP16)	Single (FP32)	Double (FP64)	Unit
Total Bits	16	32	64	bits
Sign Bits	1	1	1	bit
Exponent Bits	5	8	11	bits
Mantissa Bits	10	23	52	bits
Exponent Bias	15	127	1023	-
Decimal Digits	~3.3	~7.2	~15.9	digits
Max Value	6.55×10^{4}	3.40×10^{38}	1.80×10^{308}	-
Min Positive	6.10×10^{-5}	1.18×10^{-38}	2.23×10^{-308}	-

3. Half Precision (FP16)

Half precision floating point, also known as binary16, uses 16 bits of storage. It was introduced primarily for graphics processing and machine learning applications where memory bandwidth is critical and reduced precision is acceptable.

Bit Layout

Sign (1 bit)	Exponent (5 bits)	Mantissa (10 bits)
`Bit 15`	`Bits 14-10`	`Bits 9-0`

Example: Representing 3.14

Step 1: Convert 3.14 to binary: $3.14_{10} \approx 11.001000111..._2$

Step 2: Normalize: 1.1001000111×2^1

Step 3: Calculate exponent: $E = 1 + 15 = 16 = 10000_2$

Step 4: Extract mantissa (10 bits): 1001000111_2

`Result: 0 10000 1001000111 = 0x4248`

4. Single Precision (FP32)

Single precision floating point, also known as binary32 or simply "float" in many programming languages, uses 32 bits of storage. It provides approximately 7 decimal digits of precision and is the most commonly used format for general-purpose computing.

Bit Layout

Sign (1 bit)	Exponent (8 bits)	Mantissa (23 bits)
`Bit 31`	`Bits 30-23`	`Bits 22-0`

Example: Representing -6.625

Step 1: Sign bit = 1 (negative number)

Step 2: Convert 6.625 to binary: 110.101_2

Step 3: Normalize: 1.10101×2^2

Step 4: Calculate exponent: $E = 2 + 127 = 129 = 10000001_2$

Step 5: Extract mantissa (23 bits): $10101000000000000000000_2$

`Result: 1 10000001 10101000000000000000000 = 0xC0D40000`

5. Double Precision (FP64)

Double precision floating point, also known as binary64 or simply "double" in programming languages, uses 64 bits of storage. It provides approximately 15-16 decimal digits of precision and is the default choice for scientific computing and financial calculations where accuracy is paramount.

Bit Layout

Sign (1 bit)	Exponent (11 bits)	Mantissa (52 bits)
`Bit 63`	`Bits 62-52`	`Bits 51-0`

Example: Representing 0.1

The decimal 0.1 cannot be represented exactly in binary floating point:

Step 1: Sign bit = 0 (positive)

Step 2: $0.1_{10} = 0.0001100110011..._2$ (repeating pattern)

Step 3: Normalize: $1.1001100110011... \times 2^{-4}$

Step 4: Exponent: $E = -4 + 1023 = 1019 = 01111111011_2$

`Result: 0x3FB999999999999A (approximately)`

Note: This demonstrates why 0.1 + 0.2 ≠ 0.3 exactly in floating point arithmetic.

6. Special Values

IEEE 754 defines several special values to handle edge cases:

Value	Sign	Exponent	Mantissa
+0	0	All 0s	All 0s
-0	1	All 0s	All 0s
+∞	0	All 1s	All 0s
-∞	1	All 1s	All 0s
NaN	0 or 1	All 1s	Non-zero
Denormalized	0 or 1	All 0s	Non-zero

7. Practical Use Cases

Half Precision (FP16)

- Machine learning inference and training (reducing memory footprint)
- Graphics processing and HDR image formats

- Neural network weights where precision is less critical

Single Precision (FP32)

- General-purpose scientific computing
- 3D graphics and game development
- Signal processing applications

Double Precision (FP64)

- Financial calculations requiring high accuracy
- Scientific simulations (climate modeling, physics)
- Cryptographic computations
- Precision-critical engineering applications

Reference based on IEEE 754-2019 Standard for Floating-Point Arithmetic