Understanding IEEE 754 Floating Point

1. Introduction

Floating-point numbers let computers represent very large, very small, and fractional values using a fixed number of bits. IEEE 754 is the standard that defines how these bits are divided into *sign*, *exponent*, and *mantissa*. In this lesson you'll learn:

- How the IEEE 754 format is structured and why the exponent uses a *bias*.
- Step-by-step conversion from decimal to IEEE 754 binary (16-bit) and back.
- Two detailed examples worked through each substep.
- Eight practice problems to reinforce your understanding.

2. IEEE 754 Format Overview

An IEEE 754 floating-point number is laid out as:

$$\underbrace{\text{Sign}}_{1\text{ bit}} \mid \underbrace{\text{Exponent}}_{E\text{ bits}} \mid \underbrace{\text{Mantissa}}_{M\text{ bits}}$$

- **Sign bit (S)**: 0 for positive, 1 for negative.
- **Exponent**: stored with a *bias* so that both positive and negative exponents become nonnegative integers.
- **Mantissa** (also called *fraction*): the significant digits, stored without bias, with an *implicit leading 1* in normalized numbers.

For common precisions:

Precision	Total bits	Exponent bits (E)	Mantissa bits (M)
Half (16-bit)	16	5	10
Single (32-bit)	32	8	23
Double (64-bit)	64	11	52

2.1 Why a Bias?

Exponents can be positive or negative. Instead of using a separate sign bit for the exponent, IEEE 754 adds a **bias** to make the stored exponent always nonnegative:

$$e_{\text{stored}} = e_{\text{actual}} + (2^{E-1} - 1)$$

For $E = 5$, the bias is $2^4 - 1 = 15$. For example:

e_{actual}	e_{stored}
-15	0
-14	1
$\vdots$	$\vdots$
0	15
1	16
$\vdots$	$\vdots$
15	30
16	31

Storing exponents as unsigned integers simplifies hardware comparison and sorting.

3. Converting Decimal to IEEE 754 (16-bit)

We will convert 6.75_{10} step by step.

Step 1: Sign Bit

Since 6.75 is positive, $S = 0$.

Step 2: Convert to Binary

Split into integer part (6) and fractional part (0.75).

2.1 Integer Part (repeated division by 2)

$$\begin{aligned} 6 \div 2 &= 3 \quad \text{remainder } 0 \\ 3 \div 2 &= 1 \quad \text{remainder } 1 \\ 1 \div 2 &= 0 \quad \text{remainder } 1 \end{aligned}$$

Reading the remainders **bottom to top** gives $6_{10} = 110_2$.

2.2 Fractional Part (repeated multiplication by 2)

$$\begin{aligned} 0.75 \times 2 &= 1.50 \quad \text{write 1, remainder 0.50} \\ 0.50 \times 2 &= 1.00 \quad \text{write 1, remainder 0.00} \end{aligned}$$

So $0.75_{10} = .11_2$. Combine: $6.75_{10} = 110.11_2$.

Step 3: Normalize

Move the binary point so the form is $1.xxxxx \times 2^e$:

$$110.11_2 = 1.1011_2 \times 2^2$$

We moved the point left 2 places, so $e_{\text{actual}} = 2$.

Step 4: Biased Exponent

Bias is 15, so

$$e_{\text{stored}} = e_{\text{actual}} + 15 = 2 + 15 = 17 = 10001_2$$

Step 5: Mantissa

Drop the leading 1, take the next 10 bits of 1.1011 (pad with zeros if needed): `1011000000`.

Step 6: Assemble

So the IEEE 754 16-bit representation is:

$$0 \mid 10001 \mid 1011000000$$

Example: Convert -2.625

1. $S = 1$ (negative).
2. $2.625 = 10.101_2$:
   - $2_{10} = 10_2$
   - $0.625 \times 2 = 1.25$ (write 1, remainder 0.25)
     $0.25 \times 2 = 0.5$ (write 0, remainder 0.5)
     $0.5 \times 2 = 1.0$ (write 1, remainder 0.0)
     So $0.625_{10} = .101_2$.
3. Normalize: $10.101_2 = 1.0101_2 \times 2^1$
4. Biased exponent: $e_{\text{stored}} = 1 + 15 = 16 = 10000_2$
5. Mantissa: drop the leading 1, next 10 bits: `0101000000`
6. Final: $1 \mid 10000 \mid 0101000000$

4. Converting IEEE 754 Back to Decimal

Given (S, e_{stored}, mantissa bits):

$$\begin{aligned}\text{Sign} &= (-1)^S \\ e &= e_{\text{stored}} - \text{Bias} \\ M &= 1 + \sum_{i=1}^{10} \text{bit}_i \cdot 2^{-i} \\ \text{Value} &= \text{Sign} \times M \times 2^e\end{aligned}$$

Example: 0 | 10001 | 1011000000

1. $S = 0$, so positive.
2. $e_{\text{stored}} = 17$, so $e = 17 - 15 = 2$.
3. Mantissa $= 1 + 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = 1 + 0.5 + 0 + 0.125 + 0.0625 = 1.6875$
4. Value $= 1.6875 \times 2^2 = 6.75$

Example: 1 | 10000 | 0101000000

1. $S = 1$, so negative.
2. $e_{\text{stored}} = 16$, so $e = 16 - 15 = 1$.
3. Mantissa $= 1 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 1 + 0 + 0.25 + 0 + 0.0625 = 1.3125$
4. Value $= -1.3125 \times 2^1 = -2.625$

5. Practice Problems

Convert with all steps:

1. Convert 3.125 to 16-bit IEEE 754 and back.
2. Convert -0.15625 to 16-bit IEEE 754 and back.
3. Convert 12.5 to 16-bit IEEE 754.
4. Given 0 | 01110 | 0010000000, convert to decimal.
5. Convert -7.75 to 16-bit IEEE 754.
6. Convert 0.1 to 16-bit IEEE 754 (approximate).
7. Convert 15.875 to 16-bit IEEE 754.
8. Explain why 1/10 is infinite in binary.