perplexity

why exponents were needed, why not jus 63 bits

Exponents are needed in floating-point (IEEE 754) formats because they enable representation of both extremely large and extremely small numbers—something you cannot achieve by just using all available bits for the integer (mantissa) part. If you used all 63 bits (in a 64-bit word, accounting for one sign bit), you could only represent integers between very large negative and positive values, but you would lose the ability to represent fractions or very tiny/huge numbers efficiently.

Why Not Just 63 Bits?

- Using a fixed 63-bit integer gives a range of exactly representable whole numbers (from -2^{62} to $2^{62}-1$), but it can't represent fractions or numbers like 0.1, 1/3, or scientific values like 1.5×10^{20} or 3.2×10^{-9}.
- You can't meaningfully encode very small numbers (e.g., 1×10^{-60}) or very large numbers (e.g., 1×10^{60}) compactly—essential for science, finance, and engineering—using just integers.

What Exponents Do

- The exponent lets each number be stored in "scientific notation": $ value = (-1)^{sign} \times 1.mantissa \times 2^{exponent} $[1].
- By dedicating part of the bits to the exponent:
  - **You get a "floating" binary point**, so values can "scale" up or down in magnitude.
  - A single value can efficiently represent 0.000000012, 120000000, and everything in between, with only 64 bits.
- This encoding allows a wide **dynamic range**—from around 2×10^{-308} to 1.8×10^{308} for 64-bit doubles[1].

Why Not Use All Bits as Fraction (Mantissa)?

- More mantissa bits = more precision, but less range.
- More exponent bits = more range, but less precision.
- Floating-point design is a trade-off: part for range (exponent), part for detail (mantissa)[2].

Why Not Store Negative Exponents Directly?

- Exponents are stored using a **bias** (1023 for double, 127 for single) so that both positive and negative exponents can be represented using only non-negative binary values[3] [2].

Summary Table: Fixed vs. Floating Point

Representation	Range (approx.)	Can represent fractions?	Dynamic scaling?
63-bit Integer	-2^{62} to $2^{62}-1$	No	No
64-bit Float	~10^{-308} to ~10^{308}	Yes	Yes

In summary:

- Exponents in IEEE 754 floating-point are essential for representing a huge range of both very small and very large numbers with a fixed bit size—beyond what would be possible using only a long string of integer bits[1] [3] [2].

⁂

1. https://en.wikipedia.org/wiki/Single-precision_floating-point_format
2. https://www.doc.ic.ac.uk/~eedwards/compsys/float/
3. https://www.geeksforgeeks.org/computer-organization-architecture/ieee-standard-754-floating-point-numbers/