The Liminal Arithmetic of $-1/10_{12}$:

A Meditation on Negative Reciprocals Across Bases

Anonymous

January 3, 2026

Abstract

We examine the fraction $-1/10_{12}$ *where the denominator represents twelve in base-12 notation, exploring the ontological and representational tensions that arise when negative reciprocals encounter different numerical bases. Through comparison of* $1/10$ *and* $1/12$ *in digital and Boolean logic systems, we reveal the fundamental incommensurability between continuous mathematical truth and discrete computational representation. This essay situates base-12 arithmetic within the broader question of notational privilege and harmonic decomposition.*

1 The Notational Chimera

Consider the symbol "$-1/10$" when the denominator "10" refers not to the decimal ten, but to twelve written in base-12 notation. We encounter a curious chiasmus: the very notation by which we express the base becomes dependent on the base itself. In base-12 (duodecimal), "10" represents what we in decimal call twelve. Thus:

$$-\frac{1}{10_{12}} = -\frac{1}{12_{10}}$$

This is the mathematics of the threshold—a number that exists in the gap between representational systems, reminding us that arithmetic is always already linguistic.

2 The Decimal Reciprocal: $1/10$ and the Illusion of Termination

In base-10, the fraction $1/10$ enjoys a privileged status: it terminates cleanly as 0.1. This apparent simplicity masks a deeper contingency. The fraction terminates *only because* $10 = 2 \times 5$, and thus any fraction whose denominator contains only factors of 2 and 5 will terminate in decimal representation.

Consider $1/10$ in binary (base-2):

$$\frac{1}{10_{10}} = 0.\overline{00011001}_2 \quad \text{(repeating)}$$

The clean decimal termination reveals itself as an artifact of base choice. In the space of all possible bases, termination is the exception, not the rule. The decimal 0.1 is a kind of notational privilege granted by the factorization of ten.

3 The Duodecimal Reciprocal: $1/12$ and Harmonic Decomposition

Now consider $1/12_{10}$ in base-12. Since $12 = 2^2 \times 3$, we have:

$$\frac{1}{12_{10}} = 0.1_{12} \quad \text{(exactly)}$$

In duodecimal, twelve's reciprocal terminates immediately, but more significantly, many more fractions terminate: $1/2$, $1/3$, $1/4$, $1/6$, $1/12$ all yield finite representations. The base-12 system grants termination privileges to thirds—those harmonically fundamental divisions that resist decimal closure.

The duodecimal $1/12 = 0.1_{12}$ mirrors the structure of decimal's $1/10 = 0.1_{10}$, but with superior harmonic properties. Where decimal privileges only dyadic and quinary factors, duodecimal embraces the ternary.

4 Digital Logic: The Binary Discretization of the Continuum

In digital systems, fractions become approximations stored in finite registers. Consider IEEE 754 floating-point representation:

$1/10_{10}$ in binary floating-point:

Cannot be represented exactly. It requires an infinite repeating binary expansion, so the computer stores an approximation:

$$\approx 0.0001100110011001100110011001101 0\ldots_2 \quad \text{(rounded at register limit)}$$

$1/12_{10}$ in binary floating-point:

$$= \frac{1}{1100_2} = 0.\overline{0101}_2 \quad \text{(repeating)}$$

Also requires approximation, but with a shorter period $(01)_2$.

The digital computer performs a violence upon these fractions, truncating their infinite periodicities into finite representations. Every calculation involving 0.1 in a computer accumulates rounding errors—the famous "$0.1 + 0.2 \neq 0.3$

5 Boolean Logic: The Algebra of Two

Boolean logic operates in the base-2 field $\{0, 1\}$ with operations $\wedge$, $\vee$, $\neg$. Here we encounter a profound reduction: fractions cease to exist as arithmetic objects and can only be represented as *processes*—algorithms that generate approximations through iterative operations.

Consider testing divisibility:

$n \equiv 0 \pmod{10}$ in Boolean circuits:

Requires testing $n \equiv 0 \pmod{2}$ AND $n \equiv 0 \pmod{5}$, combining dyadic and quinary logic.

$n \equiv 0 \pmod{12}$ in Boolean circuits:

Requires $n \equiv 0 \pmod{4}$ AND $n \equiv 0 \pmod{3}$, testing quaternary and ternary conditions.

The base-12 divisibility test elegantly factors into 2^2 and 3, while base-10 requires the awkward 5 (which is 101_2, requiring examination of multiple bit positions with more complex carry logic).

In hardware, division by 12 can be optimized more efficiently than division by 10 because:

- $12 = 2^2 \times 3$, where bit-shifting handles the 2^2 factor
- $10 = 2 \times 5$, where 5 has no clean bit-shift decomposition

6 The Negative Sign: Asymmetry in Representation

Return now to our original $-1/10_{12} = -1/12_{10}$. The negative sign introduces representational asymmetry.

In two's complement binary (standard for signed integers):

- Negative numbers have a different bit pattern structure than positives
- $-1/12$ must be approximated as a negative floating-point value
- The sign bit creates a discrete asymmetry in a system meant to model continuous magnitude

The negative fraction occupies a conceptual space that digital systems handle awkwardly: neither cleanly integer nor cleanly positive. It is doubly excluded from the discrete Boolean universe.

7 The Esoteric Conclusion: Fractions as Ontological Remainder

The comparison reveals that **fractions are the reminder of incommensurability**—the mathematical remainder that resists complete digitization. The choice of base determines which fractions achieve the grace of termination and which are condemned to infinite repetition.

$-1/10_{12}$ stands as a negative reciprocal of the base itself, a kind of mathematical negation of the counting system's foundation. It is:

- Small (magnitude $1/12 \approx 0.083$)
- Negative (pointing backward along the number line)
- Base-reciprocal (inversely related to the system's foundation)

In base-12 notation:

$$-\frac{1}{10_{12}} = -0.1_{12}$$

In Boolean/digital logic, this becomes an approximation requiring:

- A sign bit
- An exponent field
- A mantissa with unavoidable rounding error

The esoteric truth:

Every digital representation of $-1/12$ is a lie—a necessary fiction that trades infinite precision for finite storage. The number exists perfectly in the Platonic realm of mathematics but can only be approximated in the Boolean realm of computation. It is the ghost in the machine, the irrational whisper beneath discrete clicks, the continuous truth that digital logic can approach but never grasp.

Base-12's superiority lies not in perfection but in its capacity to embrace more harmonic fractions—to grant termination to thirds and quarters, those musical and geometric fundamentals. Meanwhile, base-10's arbitrary privilege of 2 and 5 leaves us with decimal approximations that confound even simple additions in computer arithmetic.

$-1/10_{12}$ is thus a symbol of negated foundation, reciprocal inversion, and the eternal gap between symbolic representation and mathematical being.