Converting Decimal Fractions to Binary Reals

Method 1

- Write the decimal fraction as a sum of fractions in the form (1/2 + 1/4 + 1/8 + ...).
- Translate each fraction in the sum to binary.
- Add the binary fractions together to get the final result.

Method 2

- Convert the numerator and denominator of the decimal fraction to binary.
- Perform long division on the binary numbers.
- The quotient of the division is the binary representation of the decimal fraction.

Example

Let's use the second method to convert the decimal fraction 0.5 to binary.

- Convert the numerator and denominator of the decimal fraction to binary.
- 5 in decimal is 101 in binary.
- 10 in decimal is 1010 in binary.
- Perform long division on the binary numbers:

$$
\begin{array}{r}
.1 \\
1010 \overline{)\,0101.0} \\
\underline{-1010} \\
0
\end{array}
$$

When 1010 binary is subtracted from the dividend the remainder is zero, and the division stops.

Therefore, the decimal fraction 5/10 equals 0.1 binary. We will call this approach the binary long division method.

Representing 0.2 in Binary

Let’s convert decimal 0.2 (2/10) to binary using the binary long division method. First, we divide binary 10 by binary 1010 (decimal 10).

The first quotient large enough to use is 10000. After dividing 1010 into 10000, the remainder is 110.

Appending another zero, the new dividend is 1100. After dividing 1010 into 1100, the remain- der is 10. After appending three zeros, the new dividend is 10000.

This is the same dividend we started with. From this point on, the sequence of the bits in the quotient repeats (0011. . .), so we know that an exact quotient will not be found and 0.2 cannot be represented by a finite number of bits.

The single-precision encoded significand is **00110011001100110011001.**

```
                .00110011 (etc.)
          ┌──────────────────────
1010      │ 10.000000000
          │  1010
          ├─────
            1100
            1010
            ────
              10000
               1010
               ────
                1100
                1010
                ────
                 etc.
```

==========

Examples of decimals factored as binary reals:

Decimal Fraction	Factored As...	Binary Real
1/2	1/2	.1
1/4	1/4	.01
3/4	1/2 + 1/4	.11
1/8	1/8	.001
7/8	1/2 + 1/4 + 1/8	.111
3/8	1/4 + 1/8	.011
1/16	1/16	.0001
3/16	1/8 + 1/16	.0011
5/16	1/4 + 1/16	.0101

The table above shows how to convert decimal fractions to binary reals using the first method I described above.

The first column of the table shows the decimal fraction. The second column shows the factored form of the decimal fraction.

The third column shows the binary representation of the decimal fraction.

For example, the first row of the table shows the decimal fraction 1/2. The factored form of 1/2 is 1/2. The binary representation of 1/2 is .1.

The second row of the table shows the decimal fraction 1/4. The factored form of 1/4 is 1/4. The binary representation of 1/4 is .01.

The third row of the table shows the decimal fraction 3/4. The factored form of 3/4 is 1/2 + 1/4. The binary representation of 3/4 is .11.

The rest of the rows in the table follow the same pattern.

To use the table to convert a decimal fraction to binary, simply find the decimal fraction in the first column of the table. The binary representation of the decimal fraction will be in the third column.

For example, to convert the decimal fraction 5/8 to binary, find the decimal fraction 5/8 in the first column of the table. The binary representation of 5/8 is .101.

It is important to note that not all decimal fractions can be represented exactly in binary. This is because there is an infinite number of real numbers, but only a finite number of binary digits. As a result, some decimal fractions must be approximated when they are converted to binary.

==========

The **factored column** in the table shows how the decimal fraction can be expressed as a sum of fractions in the form (1/2 + 1/4 + 1/8 + ...). This is useful because each of the fractions in the sum can be easily converted to binary.

To factor a decimal fraction, you can use the following steps:

- Find the largest power of two that is less than or equal to the decimal fraction.
- Subtract that power of two from the decimal fraction.
- Repeat steps 1 and 2 until the decimal fraction is zero.
- The factors of the decimal fraction are the powers of two that were subtracted in each step.

For example, to factor the decimal fraction 3/4, we would do the following:

- The largest power of two that is less than or equal to 3/4 is 1/2.
- Subtracting 1/2 from 3/4 gives us 1/4.
- The largest power of two that is less than or equal to 1/4 is 1/4.
- Subtracting 1/4 from 1/4 gives us zero.
- Therefore, the factors of 3/4 are 1/2 and 1/4.

Once you have factored the decimal fraction, you can convert each of the factors to binary. To do this, **simply divide the numerator of the factor by the denominator. The quotient is the binary representation of the factor.**

For example, to convert the factor 1/2 to binary, we would divide the numerator 1 by the denominator 2. The quotient is 0.5, which is the binary representation of 1/2.

To convert the factor 1/4 to binary, we would divide the numerator 1 by the denominator 4. The quotient is 0.25, which is the binary representation of 1/4.

Once you have converted each of the factors to binary, you can add them together to get the binary representation of the decimal fraction.

For example, the binary representation of 3/4 is 0.11, which is the sum of the binary representations of 1/2 and 1/4.

==========

Questions

==========

Why doesn't the single-precision real format permit an exponent of -127?

The single-precision real format does not permit an exponent of -127 because it would generate a zero. This is because the unbiased exponent is calculated by subtracting 127 from the exponent. If the exponent is -127, then the unbiased exponent would be zero. However, the significand of a non-zero floating-point number cannot be zero. Therefore, an exponent of -127 is not permitted.

Why doesn't the single-precision real format permit an exponent of 128?

The single-precision real format does not permit an exponent of 128 because it would generate an overflow. This is because the unbiased exponent is calculated by subtracting 127 from the exponent. If the exponent is 128, then the unbiased exponent would be 1. However, the unbiased exponent cannot be greater than 255. Therefore, an exponent of 128 is not permitted.

In the IEEE double-precision format, how many bits are reserved for the fractional part of the significand?

The IEEE double-precision format reserves 52 bits for the fractional part of the significand. This gives double-precision floating-point numbers a much higher degree of precision than single-precision floating-point numbers.

In the IEEE single-precision format, how many bits are reserved for the exponent?

The IEEE single-precision format reserves 8 bits for the exponent. This gives single-precision floating-point numbers a range of values from 2^{-126} to 2^{127}.