1. HAFTA

BLM323

SAYISAL ANALİZ

Okt. Yasin ORTAKCI

yasinortakci@karabuk.edu.tr
Binary Machine Numbers

A 64-bit (binary digit) representation is used for a real number. The first bit is a sign indicator, denoted $s$. This is followed by an 11-bit exponent, $c$, called the characteristic, and a 52-bit binary fraction, $f$, called the mantissa. The base for the exponent is 2. According to IEEE 754-2008 standard a 64-bit (binary digit) representation is used for a real number. The parts of a real number representation is shown in the below:

| $s$ (sign bit): 1 bit | $c$ (characteristic): 11 bit | $f$ (fraction): 52 bit |

Since 52 binary digits correspond to between 16 and 17 decimal digits, we can assume that a number represented in this system has at least 16 decimal digits of precision. The exponent of 11 binary digits gives a range of $0$ to $2^{11} - 1 = 2047$. However, using only positive integers for the exponent would not permit an adequate representation of numbers with small magnitude. To ensure that numbers with small magnitude are equally representable, 1023 is subtracted from the characteristic, so the range of the exponent is actually from $-1023$ to 1024.

To save storage and provide a unique representation for each floating-point number, a normalization is imposed. Using this system gives a floating-point number of the form:

$$(-1)^s \times 2^{c-1023} \times (1 + f)$$

The maximum value that $c$ can take is 2047.

**Example:** Consider the machine number and convert it to decimal number.

0 1000000011 101110010001000000000000000000000000000000000000000000

**Solution:**
The left most bit is $s = 0$, which indicates that the number is positive. The next 11 bits, 10000000011, give the characteristic and are equivalent to the decimal number

\[
\text{Characteristic} \ (c) = 1 \times 2^{10} + 1 \times 2^1 + 1 \times 2^0 = 1027
\]

Fraction \ (f) = 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-8} + 1 \times 2^{-12}

\[
= \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{4096} = 0.722900390625
\]

Decimal Number = $(-1)^0 \times 2^{1027-1023} \times (1 + 0.722900390625) = 27,56640625$

0 10000000011 101110010010000000000000000000000000000000000000000

The next smallest machine number is

0 10000000011 10111001000011111111111111111111111111111111

and the next largest machine number is

0 10000000011 101110010001000000000000000000000000000000000000001

This means that our original machine number represents not only 27.56640625, but also half of the real numbers that are between 27.56640625 and the next smallest machine number, as well as half the numbers between 27.56640625 and the next largest machine number. To be precise, it represents any real number in the interval

\[ [27.566406249999982236431605997495353221893310546875, 27.5664062500000017763568394002504646778106689453125] \ (27.56640625-2^{-52}, 27.56640625+2^{-52}) \].
There are endless number of reel number in this interval. So there are many reel numbers in this interval which can not be represented by using 64-bit (binary digit) representation.

The smallest normalized positive number that can be represented has \( s = 0, c = 1, \) and \( f = 0 \) and is equivalent to

\[
0 \ 0000000001 \ 0000000000000000 \ldots \ 000'\text{dir.}
\]

\[
= (-1)^0 \times 2^{1-1023} \times (1 + 0) \approx 0.22251 \times 10^{-307}, \text{dir.}
\]

and the largest has \( s = 0, c = 2047, \) and \( f = 1 - 2^{-52} \) and is equivalent to

\[
0 \ 11111111111 \ 11111111111 \ldots \ 111'\text{dir.}
\]

\[
= (-1)^0 \times 2^{2047-1023} \times (2 - 2^{-52}) \approx 0.17977 \times 10^{309}, \text{dir.}
\]

Numbers occurring in calculations that have a magnitude less than

\[
2^{1022} (1 + 0)
\]

result in underflow and are generally set to zero. Numbers greater than

\[
2^{1023} (2 - 2^{-52})
\]

result in overflow and typically cause the computations to stop (unless the program has been designed to detect this occurrence).

Note that there are two representations for the number zero; a positive 0 when \( s = 0, c = 0 \) and \( f = 0 \), and a negative 0 when \( s = 1, c = 0 \) and \( f = 0 \).

**Decimal Numbers**

Numbers with decimal part are written in a normalized floating-point form where only significant digits are stored.
\[ \pm 0, d_1d_2 \ldots \ldots d_k \ d_{k+1} \times 10^n \]

\[ 0 \leq d_i \leq 9 \text{ ve } i = 2, 3, \ldots, k \]

**Example:** \( 159,789 = 0,159789 \times 10^3 \)

The floating-point form of any real number is obtained by terminating the mantissa of the number at \( k \) decimal digits. There are two common ways of performing this termination.

**1. Rounding Method:**

If \( d_{k+1} \geq 5 \) increase \( d_k \) one more and the number transforms to:

\[ \pm 0, d_1d_2 \ldots \ldots d'_k \times 10^n \ (d'_k = d_k + 1) \]

**2. Chopping Method:**

It is to simply chop off the digits \( d_{k+1}, d_{k+2} \ldots \ldots \) and the number transforms to:

\[ \pm 0, d_1d_2 \ldots \ldots d_k \times 10^n \]

**Ex:** Determine the five-digits (a) chopping and (b) rounding values of the irrational number \( \pi \).

\( \pi = 3,14159265 \)

Written in normalized decimal form:

\( \pi = 0,314159265 \times 10^1 \)

a. By chopping \( \pi = 0,31415 \times 10^1 \)

b. By rounding \( \pi = 0,31416 \times 10^1 \)

**Source**