Representation of Numbers and their Algorithms

James E. Stine
Electrical and Computer Engineering Department
Oklahoma State University
Stillwater, OK 74078, USA

1. Algorithms are well-defined computational procedures that takes some values as input and produces some value or set of values as an output.

2. An algorithm is said to be correct for every instance (input), if it halts or stops with the correct output.

   - Reference: Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms, MIT Press

3. Analyzing an algorithm has come to mean predicting the resource that the algorithm requires.

   - Running times of an algorithm on a particular input is the number of primitive operations or steps executed and is useful so that is is machine or hardware independent as possible.

4. Order of growth of an algorithm gives a simple characteristic of the algorithm's efficiency and also allows us to compare the relative performance of alternative algorithms.

   - Studying the asymptotic efficiency of algorithms is determining how running time of an algorithm increases with size of an input in the limit as the size of the input increases without bound

5. Big O notation

   - $\Omega(f)$ function that grows at least as fast as f
   - $\Theta(f)$ function that grows at the same rate as f
   - $O(f)$ function that grows no faster than f

6. Most computer systems represent numbers using strings of binary digits.

   - Numbers often utilize weighted number systems where a set of weights are are used to represent a specific position.

   $$x = \sum_{i=0}^{n-1} x_i \cdot w_i$$

   where the weight vector is the following

   $$W = (w_{n-1}, \ldots, w_0)$$

   and the radix (r) is used within the weight.

   $$R = (r_{n-1}, \ldots, r_0)$$

   such that

   $$w_i = r^i$$

   - Weighted number systems are typically cannonical, such that:

   $$D_i = 0, 1, 2, \ldots, \mid r_i \mid -1$$

   - Conventional number systems utilized a fixed radix positive and a canonical digit set.

7. In fixed point number systems, the position of the radix point is constant.

8. The two most common fixed point representations are integer and fractional. Integer representations are commonly used on general purpose computers, whereas, digital signal processors typically use both integer and fractional representations.

9. For example, n-bit integers have the form:

   $$A = a_{n-1}a_{n-2} \ldots a_1 a_0.$$

   The most significant bit a_{n-1} is referred to as the *sign* bit. If the sign bit is one, the number is negative. Otherwise, it is positive.

10. Similarly, n-bit fractional numbers have the form:
$$A = a_{n-1}.a_{n-2}\dots a_1 a_0$$

11. Fixed Point Binary Integer

    (a) A n-bit conventional fixed-point binary integer has the following form:
    $$A = \sum_{i=0}^{n-1} a_i \cdot 2^i$$

    (b) Most-significant bit (MSB) is the bit to the furthermost left of the number.

    (c) Least-significant bit (LSB) is the bit to the furthermost right of the number.

    (d) Unit in the least-signifcant place (ulp) is r^{-m} where m is the number of digits in the fractional part and n is the number of digits in the integer part.

12. Representation of Negative Numbers

    - Sign/Magnitude System
        - $a_{n-1} = 1$ The number is not positive
        - $a_{n-1} = 0$ The number is not negative
    - Complement Representation
        - Radix Complement (e.g. two's complement)
        - Diminished-radix Complement (e.g. one's complement)

13. 2's complement numbers

    (a) The value of an n-bit 2's complement binary integer is:
    $$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i \cdot 2^i$$
    For example, with $n = 4$ and $A = 1011$.
    $$\begin{aligned} A &= -1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= -8 + 2 + 1 \\ &= -5 \end{aligned}$$

    (b) The value of an n-bit 2's complement binary fraction is:
    $$A = -a_{n-1} + \sum_{i=0}^{n-2} a_i \cdot 2^{i-n+1}$$
    For example, with $n = 4$ and $A = 1.011$
    $$\begin{aligned} A &= -1 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ &= -1 + 1/4 + 1/8 \\ &= -5/8 \end{aligned}$$

    (c) You can also get the value an n-bit binary fraction by computing the value of the n-bit binary integer and dividing by 2^{n-1}.

    (d) To negate 2's complement numbers, invert all bits and add a unit in the least position or ulp to the least significant bit. For example,
    $$\begin{array}{rcl} 3/8 & = & 0.011 \\ \textit{invert} & & 1.100 \\ \textit{increment} & + & 0.001 \\ & = & 1.101 = -3/8 \end{array}$$

14. Sign-magnitude numbers

    (a) The value of a sign-magnitude binary fraction is:
    $$A = (1 - 2 \cdot a_{n-1}) \cdot \sum_{i=0}^{n-2} a_i \cdot 2^{i-n+1}$$
    For example, with $n = 4$ and $A = 1.011$
    $$\begin{aligned} A &= (1 - 2 \cdot 1) \cdot (0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}) \\ &= -1 \cdot (1/4 + 1/8) \\ &= -3/8 \end{aligned}$$

    (b) To negate sign-magnitude numbers, invert the sign bit (i.e., the most significant bit). For example,
    $$\begin{array}{rcl} 3/8 & = & 0.011 \\ \textit{invert\_sign} & = & 1.011 = -3/8 \end{array}$$

15. 1's complement numbers

    (a) The value of a 1's complement binary fraction is:
    $$A = \sum_{i=0}^{n-2} (a_i - a_{n-1}) \cdot 2^{i-n+1}$$
    For example, with $n = 4$ and $A = 1.011$
    $$\begin{aligned} A &= (0 - 1) \cdot 2^{-1} + (1 - 1) \cdot 2^{-2} + (1 - 1) \cdot 2^{-3} \\ &= -1/2 \end{aligned}$$

    (b) To negate 1's complement numbers, invert all bits. For example,
    $$\begin{array}{rcl} 3/8 & = & 0.011 \\ \textit{invert} & & 1.100 = -3/8 \end{array}$$

16. Characteristics of Fixed Point Number Systems (See Table 1).

    (a) Positive numbers are identical for all three number systems.

    (b) The sign-magnitude and 1's complement number systems have both $+0$ and -0.

    (c) Only the 2's complement number system has a representation for -1.

17. Uses of Fixed Point Number Systems

    (a) The 2's complement number system is the most common format for fixed-point numbers (all operations are fairly easy)

    (b) The sign-magnitude number system is the most common format for floating-point numbers (lets sign be handled separately and provides both +0 and -0)

    (c) The 1's complement number system is not frequently used

18. Behavior under truncation

    (a) 2's complement number truncate toward $-\infty$. The number either decreases in value or stays the same. For example,

$$\begin{aligned} 0.011 &= 3/8 \\ 0.01 &= 2/8 = 1/4 \\ 1.011 &= -5/8 \\ 1.01 &= -6/8 = -3/4 \end{aligned}$$

    (b) Sign-magnitude numbers truncate toward 0. Positive numbers either decrease in value or stays the same, but negative numbers either increase in value or stay the same. For example,

$$\begin{aligned} 0.011 &= 3/8 \\ 0.01 &= 2/8 = 1/4 \\ 1.011 &= -3/8 \\ 1.01 &= -2/8 = -1/4 \end{aligned}$$

    (c) 1's complement numbers also truncate toward 0. For example,

$$\begin{aligned} 0.011 &= 3/8 \\ 0.01 &= 2/8 = 1/4 \\ 1.011 &= -1/2 \\ 1.01 &= -1/2 = -4/8 \\ 1.010 &= -5/8 \\ 1.01 &= -1/2 = -4/8 \end{aligned}$$

Number	2's Complement	Sign Magnitude	1's Complement
+7/8	0.111	0.111	0.111
+3/4	0.110	0.110	0.110
+5/8	0.101	0.101	0.101
+1/2	0.100	0.100	0.100
+3/8	0.011	0.011	0.011
+1/4	0.010	0.010	0.010
+1/8	0.001	0.001	0.001
+0	0.000	0.000	0.000
-0	N/A	1.000	1.111
-1/8	1.111	1.001	1.110
-1/4	1.110	1.010	1.101
-3/8	1.101	1.011	1.100
-1/2	1.100	1.100	1.011
-5/8	1.011	1.101	1.010
-3/4	1.010	1.110	1.001
-7/8	1.001	1.111	1.000
-1	1.000	N/A	N/A

Table 1: 4-bit Fixed Point Fractions