Introduction to number system and its computer presentation

Jamil Khatib

July 21, 2000

1 Introduction

From the early days of human civilization, people start counting things using their fingers, sticks or any thing. Later on, number of things they count becomes more and more, so they found the inefficiency of this system. They introduced a new system that is based weighted number system where group of sticks are replaced by single special stick.
Our counting system now which is called decimal system is based on the same idea where we have 10 different symbols or digits for numbers and set of weights (1, 10, 100, 1000, etc.) The base number is called the Radix. In past, different civilizations used different radixes, the Egyptians knew the radix 2, the Babylonians used the radix 60, Mayans used 18 and 20. The best suitable radix for computers because they have only two symbols which is easily represented in digital circuits as 0's and 1's.

2 Binary system

Weighted number systems are based on equation (1)

$$N = \sum_{M}^{j=0} b_j B^j \tag{1}$$

Where

N:	The Number
M:	Number of digits
b:	The digit
B:	System's Radix

and so the binary system is defined by $N = \sum_{M}^{j=0} b_j 2^j$

Digital systems uses the binary system to present numbers using two levels of voltages that present 1 or 0. Each digit in computer presentation of the binary system is called a *Bit*. Computers store numbers in registers that can hold fixed numbers of digits. This limits the maximum number computers can handle, for example 8-bit registers can store up to $2^8 = 256$.

3 Integers representations

Integer or fixed point numbers describe whole numbers. There are many ways to represent these numbers using the binary system.

3.1 Unsigned integers representation

It is a straightforward mapping between bits and unsigned integer numbers. So 4 bit system maps 0000-1111 to 0 to 15. The disadvantage of this representation that it lacks to the negative numbers.

3.2 Signed-magnitude representation

In this representation, the most left digit denotes the sign of the number. This representation is not commonly used in computers because it has two representations of zero (e.g. in 4-bit numbers "1000" and "0000" represent 0). The subtraction operation is no easy to be done by computers. The most important use of it is in ADC and DAC devices.

3.3 1's complement signed representation

It is the same concept as the signed-magnitude representation but the negative numbers are complemented by the 1's complement. Unsigned 1's complement

3.4 2's complement signed representation

It is the same representation as the 1's complement but instead of calculating the 1's complement calculate 2's complement. In this representation the zero has a single representation by "0000". The arithmetic subtraction is simple, that is done by adding the numbers including the sign bit after complementing the subtrahend. The problem of this representation is that the numbers are not balanced around the zero.

3.5 Examples

Unsigned		Signed-magnitude		1's complement		2's complement	
15	1111	7	0111	7	0111	7	0111
14	1110	6	0110	6	0110	6	0110
13	1101	5	0101	5	0101	5	0101
12	1100	4	0100	4	0100	4	0100
11	1011	3	0011	3	0011	3	0011
10	1010	2	0010	2	0010	2	0010
9	1001	1	0001	1	0001	1	0001
8	1000	0	0000	0	0000	0	0000
7	0111	0	1000	0	1111	-1	1111
6	0110	-1	1001	-1	1110	-2	1110
5	0101	-2	1010	-2	1101	-3	1101
4	0100	-3	1011	-3	1100	-4	1100
3	0011	-4	1100	-4	1011	-5	1011
2	0010	-5	1101	-5	1010	-6	1010
1	0001	-6	1110	-6	1001	-7	1001
0	0000	-7	1111	-7	1000	-8	1000

3.6 Other representations

There are many other representations for integer numbers but they are not commonly used and they are used for specific problems.

4 Fractional numbers

In many mathmetical operations we use values that are less than one but larger than zero. This kind of numbers we call them real or fractional numbers. These numbers are so important in our life to describe small values and ratios such as some natural constants π , e . . . .

4.1 Fractional 2's complement

Since numbers are unbounded while computer storage are, overflow may occur may occur during arithmetic operation on the bounded system. So we need a special representation to reduce the effect of this boundness, and this is where the fractional 2's complement representation came from. This representation is based on equation 2.

$$N = -x_0 + \sum_{j=1}^{B-1} x_j 2^{-j} \tag{2}$$

Where
N: The Number
B: Number of digits
x: The digit, 0 or 1

- The sign bit is represented by x_0 where 1 indicates negative sign.
- The binary point follows x_0.
- The rest of the bits, represent a positive component in decreasing order of B-1 binary weighted fractions.

Advantages of this representation:

- Zero has a single representation

4.2 Fixed Point representation

In this representation the bits are divided into two groups, one for the integer part and the other for the fractional part. "IIII.FFFF".

Advantages of this representation: This representation gives an easy way for represnting the freactional numbers but there should a good balance between the precision of the fractions and the maximum number that can be represented. Most of the time there will be waste of some bits eaither in the fractional or integer ones.

4.3 Floating Point representations

Not the same as the fixed point representation, the location of the fractionan point can be moved from one location to another according to the precision. "thats where the name came from". One importance for this representation is the prevention of overflow in mathematical operations because the precision can be changed easily. It can also increase the range of the numbers that the computer can handle because of the use of the multiplication factor "to be explained later".
The floating point representation is set of representations but all take the general format as
Exp. 0. Fraction

4.4 Exponent-Fraction representation

This representation is a direct mapping from the scientific representation that takes the form $r^e * m$. Where r is the radix, e is the exponent and m is mantissa.
In computer representation r is 2 for binary representation and the location

of the fraction point is assumed to be fixed so to change its location in the number the exponent must be changed. So only the exponent, the mantissa and the sign need to be stored in hardware.
Zero is represented by all zeros in the exponent and the mantissa bits.

4.5 Biased Exponents

In the biased exponent format, the exponent is not used as a signed number -because the sign needs extra bit- it is biased by a base number so the exponent is subtracted from the base. This means when the exponent field has the value X the number has the exponent X - Base. This biasing makes all internal exponent calculations as positive only while keeping the whole range of the exponent intact.

4.6 IEEE-754 representation

The IEEE organization defined in the IEEE-754 standard a representation of the floating point numbers and its operations.

	Single	Double	Double-Extended	Quad-Precision
Exponent(max)	+127	+1023	+16383	+16383
Exponent(min)	-126	-1022	-16382	-16382
Exponent Bias	+127	+1023	+16383	+16383
Precision(#bits)	24	53	64	113
Total Bits	32	64	80	128
Sign bits	1	1	1	1
Exp Bits	8	11	15	15
Fraction	23	52	64	112

References

[1] The rule of distributed arithmetic in FPGA-based signal processing "http://www.xilinx.com"

[2] The scientist and engineer's guide to Digital Signal Processing by Steven Smith "http://www."

[3] Introduction to Floating point calculations and IEEE 754 standard "http://www.geocities.com/SiliconValley/Pines/6639/docs/fp_summary.html"