Computer Organization
(Number representation & Arithmetics)

KR Chowdhary
Professor & Head
Email: kr.chowdhary@gmail.com
webpage: krchowdhary.com

Department of Computer Science and Engineering
MBM Engineering College, Jodhpur

November 14, 2013

Number Formats

- Factors for selection of number representation:
  1. Types of numbers to be represented: integers, real numbers, complex numbers
  2. Range of values to be encountered
  3. Precision of numbers required
  4. Cost of hardware required to store and process the numbers.
- Formats: Fixed point and floating point format. First have small range and simple hardware requirements. Second has complex hardware and large range.

$-1101.0101_2 = -13.3125_{10}$,
$\because\ 0.0101_2 = \frac{1}{2} \times 0 + \frac{1}{4} \times 1 + \frac{1}{8} \times 0 + \frac{1}{16} \times 1 = 0.3125_{10}$

Non-negative integer representation:

8 bit binary represents $0_{10} \dots 255_{10}$. If we consider that the binary number is $A = a_{n-1}a_{n-1}\dots a_0$, then,

$$A = \sum_{i=0}^{n-1} 2^i a_i$$

Signed magnitude representation:

First bit $= 0 \Rightarrow +ve$, and $1 \Rightarrow -ve$. $\therefore\ +18 = 00010010$, in 8-bit representation, and $-18 = 10010010$

$$A = \sum_{i=0}^{n-2} 2^i a_i, \text{ if } a_{n-1} = 0$$

and

$$A = -\sum_{i=0}^{n-2} 2^i a_i, \text{ if } a_{n-1} = 1.$$

Negative Number's Representation

Drawback of above notation:

1) require consideration of sign and magnitude, 2) there is double notation for 0

$+0_{10} = 00000000$, and $-0_{10} = 10000000$, hence this notation is never used.

Two's Complement representation:

- -ve numbers are represented as two's complement.
- Most significant bit is used as sign bit, while other bits are treated differently. For *n*-bit binary number, the range is -2^{n-1} to $+2^{n-1} - 1$.
- Number of representations of zero is **one** only.
- Addition and subtractions are straight forward. If the result is -ve, then actual result's magnitude is found by obtaining its two's complement.
- Boundary numbers are complement of each other, i.e., -2^{n-1} and $+2^{n-1} - 1$.

Sign magnitude v/s two's complement

Decimal Representation	Sign Magnitude	Two's Complement
+7	0111	0111
+6	0110	0110
+5	0101	0101
+4	0100	0100
+3	0011	0011
+2	0010	0010
+1	0001	0001
+0	0000	0000
-0	1000	0000
-1	1001	1111
-2	1010	1110
-3	1011	1101
-4	1100	1100
-5	1101	1011
-6	1110	1010
-7	1111	1001
-8	–	1000

IEEE754 Floating-point Format

A decimal number 9760000 can be represented in floating point as 0.97×10^7, and a fractions 0.0000976 can be represented as 0.976×10^{-4}.

$$\pm S \times B^{\pm E}$$

here S is called significand (i.e., mantissa), B is base: 10 for decimal, 2 for binary.

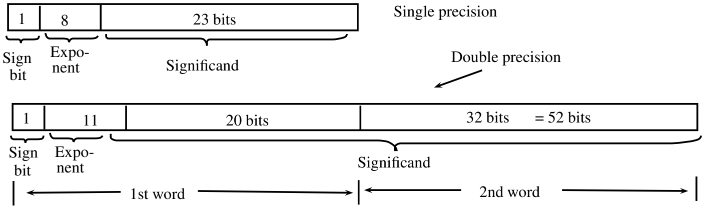


Figure 1: IEEE754 Floating-point Formats

Meaning: $(-1)^{sign} \times (1 + mantissa) \times 2^{expo.-bias}$

IEEE754 Floating-point Format

Biased Representation:

- Twos complement exponent is not effective for sorting(e.g. -1 = 1...11, 1=0 ...01). A bias is introduced so that 0...0 is smallest representable exponent, and 1 ...1 is largest.
- For k-bit exponent, a bias value of $2^{k-1} - 1$ is subtracted from the exponent to obtain the true value of exponent.
- For single precision, bias = 127, for double precision it is 1023.
- Range: SP: -2×10^{-38} to 2×10^{38}, DP: -2×10^{-308} to 2×10^{308}

Fraction:

- in scientific notations, there is no leading 0, hence an implicit leading 1 is taken in mantissa.
- Significand is taken as 24-bits, the most significant bit is always taken as 1. Thus, 32-bit biased format is

$$\pm 1.bbbb \cdots \times 2^{\pm E}$$

where b are binary, 0 and 1.

IEEE754 Floating-point Format

- Let 32-bit number in biased exponent format is: 10110101110100010000000000000000. This number is -1.1010001×2^x, where $x = 01101011 = 107_{10}$, true value of exponent is $x + 2's$ complement of 127 $(=10000001_2) =$ -00010100=-20. $\therefore$ true value of above FP representation is $-1.6328125 \times 2^{-20}$
- 23 bits mantissa provides precision equivalent to 8 decimal digits, and 53 bit to 16 digits.
- Extended single precision and double precision:
  - Exponent $= 0$, mantissa $= 0$: exact zero. With sign bit, it is ± 0
  - Exponent of all 1's and mantissa 0: ∞ is represented (over-flow due to division by 0). With sign bit it is$\pm\infty$.
  - Exponent $= 0$, nonzero mantissa is *denormal* numbers. M is non-zero 23 bit number. Value is $\pm 0.M \times 2^{-126}$. They are smaller than smallest normal no.
  - $E = 255$, $m \neq 0$: Nan(Not a number format) indicator, e.g. due to operation of 0/0 or $\sqrt{-1}$.
- Exception handling?

Floating Point Addition/subtraction

Algorithm-to-add

1. Align the two numbers by right shifting the smaller number until it matches the bigger one
2. set exponents of both equal to larger
3. Add/subtract significands (mantissas)
4. Normalize the sum
5. Check for overflow or underflow and raise an exception if necessary
6. If not normalized repeat from 3

Floating Point Multiplication

Algorithm-to-multiply

1. Add the exponents and subtract bias
2. Multiply significands
3. Normalize if necessary
4. Check for over-/underflow and raise exception if needed
5. Round significand
6. If not normalized repeat from step 3
7. Set the sign to positive if input signs are equal, negative if they differ

How to divide?

Faster Division

Cannot perform unrolling due to the strict dependency between stages, so

1. "Guess" more than one bit at a time using a precomputed Lookup Table with inputs from Remainder and Divisor
2. Do not add back the Divisor if the remainder is negative. Instead, add the Dividend to the shifted Remainder in the next step (nonrestoring division)
3. Do not store the result of the subtraction if it is negative (nonperforming division)

Floating-point Arithmetic

Arithmetic:

- Let $X = X_s \times B^{X_E}$, and $Y = Y_s \times B^{Y_E}$
- Then, $X + Y = (X_S \times B^{X_E - Y_E} + Y_S) \times B^{Y_E}$, if $X_E \leq Y_E$
  $X - Y = (X_S - Y_S \times B^{Y_E - X_E}) \times B^{X_E}$, if $X_E \geq Y_E$
  $X \times Y = (X_S \times Y_S) \times B^{X_E + Y_E}$
  $X \div Y = \frac{X_S}{Y_S} B^{X_E - Y_E}$
- Let $X = 0.3 \times 10^2 = 30$ and $Y = 0.2 \times 10^3 = 200$. Then, $X + Y = (0.3 \times 10^{2-3} + 0.2) \times 10^3 = (0.03 + 0.2) \times 10^3 = (0.23) \times 10^3 = 230$.

ASCII and Extended Binary Coded Decimal Interchg Code

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Figure 2: ASCII-table

- ASCII encodes 128 specified characters-numbers 0-9, letters a-z, A-Z, some punctuation symbols, some control codes originated with Teletype machines, a blank space, into 7-bit binary integers.
- ASCII represent text in computers, communications equipment, and other devices that use text.
- EBCDIC is 8-bit character encoding used mainly on IBM mainframe and IBM midrange systems

Practice Exercises

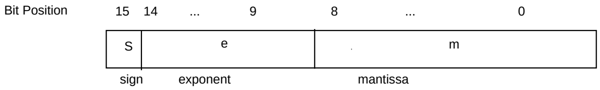
1. Express the decimal 0.5, -0.123 as signed 6-bit fractions.
2. What is the maximum representation error, e, if only 8 significant bits after decimal point are used?
3. Assuming a 6-bit exponent, 9-bit normalized fractional mantissa, and exponent is represented in biased format, add the number below: $A = 0\ 100001\ 111111110$, $B = 0\ 011111\ 0010110101$. Assume an implicit 1 to the left of mantissa.
4. Assuming all numbers are in 2's complement representation, which of the following numbers is divisible by 11111011?
   (A) 11100111 (B) 11100100 (C) 11010111 (D) 11011011

Practice Exercises

5. The hexadecimal representation of 657_8 is:
   (a) 1AF (b) D78 (c) D71 (d) 32F
6. What is answer for $(1+1\times10^{20})-(1\times10^{20})$? Justify.
7. What is answer for $1+(1\times10^{20}-1\times10^{20})$? Justify.
8. How the rounding/truncation is carried out by the CPU?
9. How overflow can be detected, if the carry bit is not used? I.e., decide it only based on the values of A, B, $C=A+B$ and $C=A-B$. Assume that two's complement is used for negative numbers.
10. The range of integers that can be represented by an n bit 2's complement number system is:
    (a) -2^{n-1} to $(2^{n-1}-1)$ (b) $-(2^{n-1}-1)$ to $(2^{n-1}-1)$
    (c) -2^{n-1} to 2^{n-1} (d) $-(2^{n-1}+1)$ to $(2^{n-1}-1)$

Practice Exercises

11. The following is a scheme for floating point number representation using 16 bits.



Let s, e, and m be the numbers represented in binary in the sign, exponent, and mantissa fields, respectively. Then the floating point number represented is: $(-1)^s(1+m\times 2^{-9})2^{e-31}$, if the number $\neq 111111$, and 0 otherwise.

What is the maximum difference between two successive real numbers representable in this system?

(A) 2^{-40} (B) 2^{-9} (C) 2^{22} (D) 2^{31}

12. What are the values of expressions $\infty/0, 0/\infty, \infty/\infty$. Justify your answer.