Floating point Numbers

Computer Mathematics I

Jiraporn Pooksook
Department of Electrical and Computer Engineering
Naresuan University

FIXED POINT NOTATION

In Fixed Point Notation, the number is stored as a signed integer in two's complement format.
The radix point is the separator between integer and fractional parts.

signed integer . fractional

0001.0110 = +
$0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4}$
= 1+ 1/4 + 1/8 = 1.375
1111.0000 => (110 + 001 = 111) =
$0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 0 \times 2^{-4}$
= -1
1100.0100 => (011 + 001 = 100) =
$1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 0 \times 2^{-4}$
= -4.25

FLOATING POINT NOTATION

Floating Point Representation is based on Scientific Notation.

- Significant (mantissa)
- Base
- Exponent

It is written in a form:

$$+/-\, mantissa \times Base^{Exponent}$$

Example:

$123.45 = 1.2345 \times 10^{2}$

$= 12.345 \times 10^{1}$

$= 1234.5 \times 10^{-1}$

NORMALISED SCIENTIFIC NOTATION

We choose an exponent so that the absolute value of the mantissa remains greater than or equal to 1 but less than the number base.

Example:
$500.0 = 5.0 \times 10^2$
$10.1_2 = 1.01 \times 2^1$
$0.111_2 = 1.11 \times 2^{-1}$

A binary number can be expressed in scientific scientific notation in several ways like decimal numbers.
$0.110010 \times 2^5 = 0.78125 \times 32 = 25$
$1.10010 \times 2^4 = 1.5625 \times 16 = 25$
$11.0010 \times 2^3 = 3.125 \times 8 = 25$
$110.010 \times 2^2 = 6.25 \times 4 = 25$
$1100.10 \times 2^1 = 12.5 \times 2 = 25$
$11001.0 \times 2^0 = 25 \times 1 = 25$

Fixed point vs Floating point

Fixed point allows calculations over a wide range of magnitudes.
Example:
$0.22 \times 0.22 = 0.0484$

Fixed point:
$0.220 \times 0.220 = 0.048$

Floating point:
$(2.2{\times}10^{-}1) \times (2.2{\times}10^{-}1) = 4.84{\times}10^{-}2$

IEEE FLOATING-POINT REPRESENTATION

Floating point is represented in a form:

$$V = (-1)^s \times M \times 2^E$$

- The sign s determines whether the number is negative (s = 1) or positive (s = 0).
- The significand M is a fractional binary number that M is either 1≤M≤2 or 0≤M≤1.
- The exponent E weights the value by a (possibly negative) power of 2 and it is stored in the two's complement format.
- The bit representation of a floating point number is divided into three fields to encode these values:
  - the single sign bit s directly code the sign s.
  - the k-bit exponent field exp = $e_{k-1} \ldots e_1 e_0$ encodes the exponent E.
  - the n-bit fraction field frac= $f_{n-1} \ldots f_1 f_0$ encodes the significand M, but the value encoded also depends on whether or not the exponent field equals 0.

IEEE FLOATING-POINT REPRESENTATION

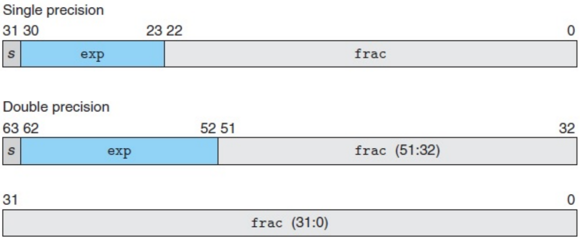

Figure 2.31 **Standard floating-point formats.** Floating-point numbers are represented by three fields. For the two most common formats, these are packed in 32-bit (single precision) or 64-bit (double precision) words.

Figure: Retrieved from Computer systems : a programmer's perspective / Randal E. Bryant, David R. O'Hallaron.-2nd ed.

IEEE FLOATING-POINT REPRESENTATION

The value encoded by a given bit representation can be divided into three different cases (the latter having two variants), depending on the value of exp.

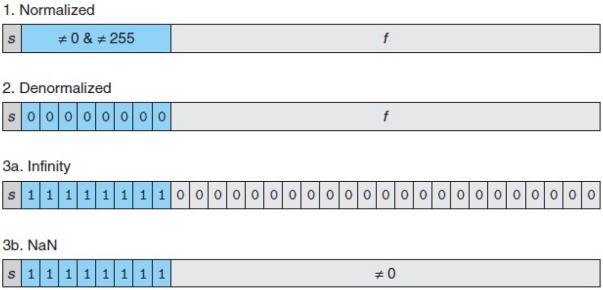

Figure 2.32 **Categories of single-precision, floating-point values.** The value of the exponent determines whether the number is (1) normalized, (2) denormalized, or a (3) special value.

Figure: Retrieved from Computer systems : a programmer's perspective / Randal E. Bryant, David R. O'Hallaron.-2nd ed.

IEEE FLOATING-POINT REPRESENTATION

Case 1: Normalized values

It occurs when the bit pattern of exp is neither all zeros nor all ones. In this case, the exponent field is interpreted as representing a signed integer in biased form.

- E = e - Bias where e is the unsigned number having bit representation $e_{k-1} \ldots e_1 e_0$, and Bias is a bias value equal to $2^{k-1} - 1$
- 0≤f≤1, having binary representation $0.f_{n-1} \ldots f_1 f_0$
- M = 1 + f so M is in the range [1,2)

Example: 8-bit floating-point format, k = 4 exponent bits n = 3 fraction bits, the bias is 7.

bit representation= 0 0110 110

- e = 6
- E = 6-7 = -1
- f = 6/8
- M = 1 + f = 14/8
- $V = M \times 2^E = 14/8 \times 1/2 = 0.875$

IEEE FLOATING-POINT REPRESENTATION

Case 2: Denormalized values
It occurs when the exponent field is all zeros. In this case, the exponent value is E = 1-Bias, and the significand value is M = f.

- E = 1 - Bias where Bias is a bias value equal to $2^{k-1} - 1$
- 0≤f≤1, having binary representation $0.f_{n-1} \ldots f_1 f_0$
- M = f

Example: 8-bit floating-point format, k = 4 exponent bits
n = 3 fraction bits, the bias is 7.
bit representation= 0 0000 000

- e = 0
- E = 1-7 = -6
- f = 0/8
- M = f = 0/8
- V = M $\times 2^E$ = $0/8 \times 1/64$ = 0.0

IEEE FLOATING-POINT REPRESENTATION

Case 3: Special values
It occurs when the exponent field is all ones.

- when the fraction field is all zeros, the resulting values represent infinity, either $+\infty$ when s=0 or $-\infty$ when s=1.
- when the fraction field is nonzero, the resulting value is called a "NaN," short for "Not a Number." Such values are returned as the result of an operation where the result cannot be given as a real number or as infinity.

IEEE FLOATING-POINT REPRESENTATION

		Exponent			Fraction		Value		
Description	Bit representation	e	E	2^E	f	M	$2^E \times M$	V	Decimal
Zero	0 0000 000	0	−6	$\frac{1}{64}$	$\frac{0}{8}$	$\frac{0}{8}$	$\frac{0}{512}$	0	0.0
Smallest pos.	0 0000 001	0	−6	$\frac{1}{64}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{512}$	$\frac{1}{512}$	0.001953
	0 0000 010	0	−6	$\frac{1}{64}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{2}{512}$	$\frac{1}{256}$	0.003906
	0 0000 011	0	−6	$\frac{1}{64}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{512}$	$\frac{3}{512}$	0.005859
	⋮								
Largest denorm.	0 0000 111	0	−6	$\frac{1}{64}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{512}$	$\frac{7}{512}$	0.013672
Smallest norm.	0 0001 000	1	−6	$\frac{1}{64}$	$\frac{0}{8}$	$\frac{8}{8}$	$\frac{8}{512}$	$\frac{1}{64}$	0.015625
	0 0001 001	1	−6	$\frac{1}{64}$	$\frac{1}{8}$	$\frac{9}{8}$	$\frac{9}{512}$	$\frac{9}{512}$	0.017578
	⋮								
	0 0110 110	6	−1	$\frac{1}{2}$	$\frac{6}{8}$	$\frac{14}{8}$	$\frac{14}{16}$	$\frac{7}{8}$	0.875
	0 0110 111	6	−1	$\frac{1}{2}$	$\frac{7}{8}$	$\frac{15}{8}$	$\frac{15}{16}$	$\frac{15}{16}$	0.9375
One	0 0111 000	7	0	1	$\frac{0}{8}$	$\frac{8}{8}$	$\frac{8}{8}$	1	1.0
	0 0111 001	7	0	1	$\frac{1}{8}$	$\frac{9}{8}$	$\frac{9}{8}$	$\frac{9}{8}$	1.125
	0 0111 010	7	0	1	$\frac{2}{8}$	$\frac{10}{8}$	$\frac{10}{8}$	$\frac{5}{4}$	1.25
	⋮								
	0 1110 110	14	7	128	$\frac{6}{8}$	$\frac{14}{8}$	$\frac{1792}{8}$	224	224.0

Figure: Retrieved from Computer systems : a programmer's perspective / Randal E. Bryant, David R. O'Hallaron.-2nd ed.

IEEE 754 STANDARD FLOATING-POINT 32 BITS

Floating point is represented in a form:

$$V = (-1)^s \times (1.f)_2 \times 2^{exponent-127}$$

- Sign for 1 bit is allocated where s determines negative (s = 1) or positive (s = 0).
- Mantissa or significand is allocated 23 bits.
- Exponent is allocated 8 bits where the value of bias is 127. Thus a stored value -127 means that exponent = 0.

b_0	$b_1\ b_2\ b_3 \ldots\ldots\ldots\ldots\ b_8$	$b_9\ b_{10}\ b_{11} \ldots\ldots\ldots\ldots\ldots\ldots b_{30}\ b_{31}$
Sign	Exponent	Significand

Figure: Retrieved from IEEE Standard for Floating Point Numbers,V Rajaraman

IEEE 754 STANDARD FLOATING-POINT 32 BITS

Example: Represent 52.21875 in 32-bits IEEE 754 standard

- bit representation 52.21875 = 110100.00111
- 52.21875 = 1.1010000111 $\times 2^5$
- Normalized significand = .1010000111
- Exponent (e-127) = 5
- Hence e = 132

Therefore the bit representation in IEEE 754 format is

0	10000100	10100001110000000000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

Figure: Retrieved from IEEE Standard for Floating Point Numbers,V Rajaraman

IEEE 754 STANDARD FLOATING-POINT 32 BITS

Representation of zero: Zero is represented in the IEEE Standard by all 0s for the exponent and all 0s for the significand.

+0

0	00000000	00000000000000000000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

–0

1	00000000	00000000000000000000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

Figure: Retrieved from IEEE Standard for Floating Point Numbers,V Rajaraman

IEEE 754 STANDARD FLOATING-POINT 32 BITS

Representation of infinity: All 1s in the exponent field is assumed to represent infinity.

$+\infty$

0	11111111	00000000000000000000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

$-\infty$

1	11111111	00000000000000000000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

Figure: Retrieved from IEEE Standard for Floating Point Numbers,V Rajaraman

IEEE 754 STANDARD FLOATING-POINT 32 BITS

Representation of Non Numbers:

- Quiet NaN which is used when the result of an operation is not defined such as 0/0.
- Signalling Nan which is used to give an error message when an operation leads to a floating point underflow like the result of a computation is smaller than the smallest number that can be stored.

QNaN

0 or 1	11111111	00010000000000000000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

SNaN

0 or 1	11111111	10000000000000100000000
Sign 1 bit	Exponent 8 bits	Significand 23 bits

IEEE 754 STANDARD FLOATING-POINT 64 BITS

Floating point is represented in a form:

$$V = (-1)^s \times (1.f)_2 \times 2^{exponent-1023}$$

- Sign for 1 bit is allocated where s determines negative (s = 1) or positive (s = 0).
- Mantissa or significand is allocated 52 bits.
- Exponent is allocated 11 bits where the value of bias is 1023.