WIKIPEDIA

Significand

The **significand**[1] (also **mantissa**[2] or **coefficient**,[1] sometimes also **argument**, or ambiguously **fraction**[3][nb 1] or **characteristic**[4][2])[5] is part of a number in scientific notation or in floating-point representation, consisting of its significant digits. Depending on the interpretation of the exponent, the significand may represent an integer or a fraction.

Contents

Example

The number 123.45 can be represented as a decimal floating-point number with the integer 12345 as the significand and a 10^{-2} power term, also called characteristics,[6][7][8] where −2 is the exponent (and 10 is the base). Its value is given by the following arithmetic:

$$123.45 = 12345 \times 10^{-2}.$$

The same value can also be represented in normalized form with 1.2345 as the fractional coefficient, and +2 as the exponent (and 10 as the base):

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

Schmid, however, called this representation with a significand ranging between 1.0 and 10 a **modified normalized form**.[7][8]

For base 2, this 1.xxxx form is also called a **normalized significand**.

Finally, the value can be represented in the format given by the Language Independent Arithmetic standard and several programming language standards, including Ada, C, Fortran and Modula-2, as

$$123.45 = 0.12345 \times 10^{+3}.$$

Schmid called this representation with a significand ranging between 0.1 and 1.0 the **true normalized form**.[7][8]

For base 2, this 0.xxxx form is also called a **normed significand**.

Significands and the hidden bit

For a normalized number, the most significant digit is always non-zero. When working in binary, this constraint uniquely determines this digit to always be 1; as such, it does not need to be explicitly stored, being called the *hidden bit*. The significand is characterized by its width in (binary) digits, and depending on the context, the hidden bit may or may not be counted towards the width of the significand. For example, the same IEEE 754 double-precision format is commonly described as having either a 53-bit significand, including the hidden bit, or a 52-bit significand, excluding the hidden bit. IEEE 754 defines the precision p to be the number of digits in the significand, including any implicit leading bit (e.g., $p = 53$ for the double-precision format), thus in a way independent from the encoding, and the term to express what is encoded (that is, the significand without its leading bit) is *trailing significand field*.

Terminology

The term *significand* was introduced by George Forsythe and Cleve Moler in 1967[9][10][11][5] and is the word used in the IEEE standard.[12] However, in 1946 Arthur Burks used the terms *mantissa* and *characteristic* to describe the two parts of a floating-point number (Burks[6] *et al.*) and that usage remains common among computer scientists today. *Mantissa* and *characteristic* have long described the two parts of the logarithm found on tables of common logarithms. While the two meanings of *exponent* are analogous, the two meanings of *mantissa* are not equivalent. For this reason, the use of *mantissa* for *significand* is discouraged by some including the creator of the standard, William Kahan[1] and prominent computer programmer and author of *The Art of Computer Programming*, Donald E. Knuth.[4]

The confusion is because scientific notation and floating-point representation are log-linear, not logarithmic. To multiply two numbers, given their logarithms, one just adds the characteristic (integer part) and the mantissa (fractional part). By contrast, to multiply two floating-point numbers, one adds the exponent (which is logarithmic) and *multiplies* the significand (which is linear).

See also

- Mantissa (logarithm)

Notes

1. The term *fraction* is used in IEEE 754-1985 with a different meaning: it is the fractional part of the significand, i.e. the significand without its explicit or implicit leading bit.

References

1. Kahan, William Morton (2002-04-19), *Names for Standardized Floating-Point Formats* (http://www.eecs.berkeley.edu/~wkahan/ieee754status/Names.pdf) (PDF), "[…] *m* is the significand or coefficient or (wrongly) mantissa […]"

2. Gosling, John B. (1980). "6.1 Floating-Point Notation / 6.8.5 Exponent Representation". In Sumner, Frank H. (ed.). *Design of Arithmetic Units for Digital Computers*. Macmillan Computer Science Series (1 ed.). Department of Computer Science, University of Manchester, Manchester, UK: The Macmillan Press Ltd. pp. 74, 91, 137–138. ISBN 0-333-26397-9. "[…] In floating-point representation, a number *x* is represented by two signed numbers *m* and *e* such that $x = m \cdot b^{e}$ where *m* is the mantissa, *e* the exponent and *b* the base. […] The mantissa is sometimes termed the characteristic and a version of the exponent also has this title from some authors. It is hoped that the terms here will be unambiguous. […] [w]e use a[n exponent] value which is shifted by half the binary range of the number. […] This special form is sometimes referred to as a biased exponent, since it is the conventional value plus a constant. Some authors have called it a characteristic, but this term should not be used, since CDC and others use this term for the mantissa. It is also referred to as an 'excess -' representation, where, for example, - is 64 for a 7-bit exponent ($2^{7-1} = 64$). […]" (NB. Gosling does not mention the term significand at all.)
3. *English Electric KDF9: Very high speed data processing system for Commerce, Industry, Science* (http://www.ourcomputerheritage.org/KDF9_Flier.pdf) (PDF) (Product flyer). English Electric. c. 1961. Publication No. DP/103. 096320WP/RP0961. Archived (https://web.archive.org/web/20200727143037/http://www.ourcomputerheritage.org/KDF9_Flier.pdf) (PDF) from the original on 2020-07-27. Retrieved 2020-07-27.
4. Knuth, Donald E. *The Art of Computer Programming*. Vol. 2. p. 214. ISBN 0-201-89684-2. "[…] Other names are occasionally used for this purpose, notably 'characteristic' and 'mantissa'; but it is an abuse of terminology to call the fraction part a mantissa, since that term has quite a different meaning in connection with logarithms. Furthermore the English word mantissa means 'a worthless addition.' […]"
5. Savard, John J. G. (2018) [2005]. "Floating-Point Formats" (http://www.quadibloc.com/comp/cp0201.htm). *quadibloc*. A Note on Field Designations. Archived (https://web.archive.org/web/20180703001709/http://www.quadibloc.com/comp/cp0201.htm) from the original on 2018-07-16. Retrieved 2018-07-16.
6. Burks, Arthur Walter; Goldstine, Herman H.; von Neumann, John (1963) [1946]. "5.3.". In Taub, A. H. (ed.). *Preliminary discussion of the logical design of an electronic computing instrument* (https://www.cs.princeton.edu/courses/archive/fall10/cos375/Burks.pdf) (PDF). *Collected Works of John von Neumann* (Technical report, Institute for Advanced Study, Princeton, New Jersey, USA). Vol. 5. New York, USA: The Macmillan Company. p. 42. Retrieved 2016-02-07. "[…] Several of the digital computers being built or planned in this country and England are to contain a so-called "floating decimal point". This is a mechanism for expressing each word as a characteristic and a mantissa—e.g. 123.45 would be carried in the machine as (0.12345,03), where the 3 is the exponent of 10 associated with the number. […]"
7. Schmid, Hermann (1974). *Decimal Computation* (https://archive.org/details/decimalcomputati0000schm) (1 ed.). Binghamton, New York, USA: John Wiley & Sons, Inc. p. 204 (https://archive.org/details/decimalcomputati0000schm/page/204) -205. ISBN 0-471-76180-X. Retrieved 2016-01-03.
8. Schmid, Hermann (1983) [1974]. *Decimal Computation* (https://books.google.com/books?id=uEYZAQAAIAAJ) (1 (reprint) ed.). Malabar, Florida, USA: Robert E. Krieger Publishing Company. pp. 204–205. ISBN 0-89874-318-4. Retrieved 2016-01-03. (NB. At least some batches of this reprint edition were misprints with defective pages 115–146.)

9. Forsythe, George Elmer; Moler, Cleve Barry (September 1967). *Computer Solution of Linear Algebraic Systems*. Automatic Computation (1st ed.). New Jersey, USA: Prentice-Hall, Englewood Cliffs. ISBN 0-13-165779-8.
10. Sterbenz, Pat H. (1974-05-01). *Floating-Point Computation*. Prentice-Hall Series in Automatic Computation (1 ed.). Englewood Cliffs, New Jersey, USA: Prentice Hall. ISBN 0-13-322495-3.
11. Goldberg, David (March 1991). "What Every Computer Scientist Should Know About Floating-Point Arithmetic" (http://perso.ens-lyon.fr/jean-michel.muller/goldberg.pdf) (PDF). *Computing Surveys*. Xerox Palo Alto Research Center (PARC), Palo Alto, California, USA: Association for Computing Machinery, Inc. **23** (1): 7. Archived (https://web.archive.org/web/20160713044143/http://perso.ens-lyon.fr/jean-michel.muller/goldberg.pdf) (PDF) from the original on 2016-07-13. Retrieved 2016-07-13. "[…] This term was introduced by Forsythe and Moler [1967], and has generally replaced the older term *mantissa*. […]" (NB. A newer edited version can be found here: [1] (https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html))
12. *754-2019 - IEEE Standard for Floating-Point Arithmetic*. IEEE. 2019. doi:10.1109/IEEESTD.2019.8766229 (https://doi.org/10.1109%2FIEEESTD.2019.8766229). ISBN 978-1-5044-5924-2.