

An American National Standard

IEEE Standard for Radix-Independent Floating-Point Arithmetic

Sponsor

**Technical Committee on Microprocessors and Microcomputers
of the
IEEE Computer Society**

Approved March 12, 1987
Reaffirmed March 17, 1994

IEEE Standards Board

Approved September 10, 1987
Reaffirmed August 23, 1994

American National Standards Institute

© Copyright 1987 by

**The Institute of Electrical and Electronics Engineers, Inc
345 East 47th Street, New York, NY 10017, USA**

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments revived from users of the standard. Every IEEE Standard is subjected to review at least once every five years for revision or reaffirmation. When a document is more than five years old, and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
345 East 47th Street
New York, NY 10017
USA

Foreword

(This Foreword is not a part of ANSI/IEEE Std 854-1987, IEEE Standard for Radix-Independent Floating-Point Arithmetic.)

This standard is a product of the Radix-Independent Floating-Point Arithmetic Working Group of the Microprocessor Standards Subcommittee. This work was sponsored by the Technical Committee on Microprocessors and Minicomputers of the IEEE Computer Society. It generalizes ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic, to remove dependencies on radix and wordlength. The committee believes that, except for a possible conflict with the requirements in 5.6 and 7.2 that unrecognizable decimal input strings signal an exception, and in 6.3 that the sign of zero be preserved in certain conversion operations, any implementation conforming to ANSI/IEEE Std 754-1985 will also conform to this standard. In addition, the definition of *logb* has been enhanced in the Appendix, and two new functions, *conv* and *nearbyinteger*, have been added. Draft 1.0 of this standard was published to solicit public comments.¹

This standard defines a family of commercially feasible ways for new systems to perform floating-point arithmetic. Issues of retrofitting were not considered. Among the desiderata that guided the formulation of this standard were the following:

- 1) Facilitate movement of existing programs from diverse computers to those that adhere to this standard.
- 2) Enhance the capabilities and safety available to programmers who, though not expert in numerical methods, may well be attempting to produce numerically sophisticated programs. However, we recognize that utility and safety are sometimes antagonists.
- 3) Encourage experts to develop and distribute robust and efficient numerical programs that are portable, via minor editing and recompilation, onto any computer that conforms to this standard and possesses adequate capacity.
- 4) Provide direct support for
 - a) execution-time diagnosis of anomalies,
 - b) smoother handling of exceptions, and
 - c) interval arithmetic at a reasonable cost.
- 5) Provide for development of
 - a) standard elementary functions like exp and cos,
 - b) very high precision (multiword) arithmetic, and
 - c) coupling of numerical and symbolic algebraic computation.
- 6) Enable rather than preclude further refinements and extensions.

Members of the Radix-Independent Floating-Point Arithmetic Working Group who voted on versions of this proposal or participated by correspondence were as follows:

W. J. Cody, *Chair*

J. Bolstad
J. Boney
L. Breed
J. T. Coonen
J. Demmel
A. A. DuBrulle
P. J. Faillace
A. Fyfe
D. M. Gay
R. Goodman
K. Hanson
D. Hough

R. E. James, III
W. Kahan
R. Karpinski
V. Klema
E. LeBlanc
C. Lewis
Z. Liu
R. Martin
R. Mateosian
W. H. McAllister
E. McDonnell
M. Mikalajunas

K. C. Ng
K. A. Norman
J. F. Palmer
R. Pexton
F. N. Ris
J. Ryshpan
D. Stevenson
R. Stewart
T. Suyehiro
H. C. Thacher, Jr
J. W. Thomas
R. J. Wytmar

¹IEEE *Micro*, vol 4, no 4, Aug 1984.

The Chairman of the Microprocessor Standards Committee at the time of approval was James Davis.

At the time of approval, the Technical Committee on Microprocessors and Microcomputers had as its Executive Committee:

Martin Freeman, *Chair*

James Flournoy

Michael Smolin

Robert Stewart

The following persons were on the balloting committee that approved this document for submission to the IEEE Standards Board:

A. Allison
P. Ashendem
G. Baldwin
R. Baker
L. Barcnas
M. Biewer
R. Boberg
P. Borrill
C. Camp
W. J. Cody
S. Cooper
J. Davis
R. Davis
S. Diamond
W. Fischer
J. Flournoy

G. Force
M. Freeman
D. Gustavson
T. Harkaway
R. Hochsprung
D. Hough
D. James
R. James
W. Kahan
L. Kaleda
R. Karpinski
H. Kirrman
D. Kraft
G. Langdon
T. Leonard
G. Lyons

R. McLellan
K. Mondal
J. Mooney
G. Nelson
D. Ogden
T. Pittman
S. Pital
B. Shields
M. Smolin
D. Stevenson
R. Stewart
M. Teener
S. Tetrick
E. Waltz
G. White
F. Whittington

When the IEEE Standards Board approved this standard on March 12, 1987, it had the following membership:

Donald C. Fleckenstein, *Chair*

Marco W. Migliaro, *Vice Chair*

Sava I. Sherr, *Secretary*

James H. Beall
Dennis Bodson
Marshall L. Cain
James M. Daly
Stephen R. Dillon
Eugene P. Fogarty
Jay Forster
Kenneth D. Hendrix
Irvin N. Howell

Leslie R. Kerr
Jack Kinn
Irving Kolodny
Joseph L. Koepfinger*
Edward Lohse
John May
Lawrence V. McCall
L. Bruce McClung
Donald T. Michael*

L. John Rankine
John P. Riganati
Gary S. Robinson
Frank L. Rose
Robert E. Rountree
William R. Tackaberry
William B. Wilkens
Helen M. Wood

*Member emeritus

CLAUSE	PAGE
1. Scope	1
1.1 Implementation Objectives	1
1.2 Inclusions	1
1.3 Exclusions	1
2. Definitions.....	1
3. Precisions	2
3.1 Sets of Values.....	2
3.2 Basic Precisions	3
3.3 Extended Precisions	4
3.4 Combinations of Precisions	4
4. Rounding	5
4.1 Round to Nearest.....	5
4.2 Directed Roundings.....	5
4.3 Rounding Precision.....	5
5. Operations	5
5.1 Arithmetic	6
5.2 Square Root.....	6
5.3 Floating-Point Precision Conversions.....	6
5.4 Conversion Between Floating Point and Integer	6
5.5 Round Floating-Point Number to Integral Value.....	6
5.6 Floating-PointDecimal String Conversion.....	6
5.7 Comparison	8
6. Infinity, NaNs, and Signed Zero	10
6.1 Infinity Arithmetic	10
6.2 Operations with NaNs	10
6.3 The Algebraic Sign	10
7. Exceptions.....	11
7.1 Invalid Operation	11
7.2 Division by Zero	11
7.3 Overflow	11
7.4 Underflow	12
7.5 Inexact.....	12
8. Traps.....	13
8.1 Trap Handler	13
8.2 Precedence	13
Annex (Informative) Recommended Functions and Predicates	14

IEEE Standard for Radix-Independent Floating-Point Arithmetic

1. Scope

1.1 Implementation Objectives

It is intended that an implementation of a floating-point system conforming to this standard can be realized entirely in software, entirely in hardware, or in any combination of software and hardware. It is the environment the programmer or user of the system sees that conforms or fails to conform to this standard. Hardware components that require software support to conform shall not be said to conform apart from such software.

1.2 Inclusions

This standard specifies the following:

- 1) Constraints on parameters defining values of basic and extended floating-point numbers
- 2) Add, subtract, multiply, divide, square root, remainder and compare operations
- 3) Conversions between integers and floating-point numbers
- 4) Conversions between different floating-point precisions
- 5) Conversion between basic precision floating-point numbers and decimal strings
- 6) Floating-point exceptions and their handling, including nonnumbers (NaNs)

1.3 Exclusions

This standard does not specify the following:

- 1) Formats for internal storage of floating-point numbers
- 2) Encodings of integers and formats of strings of characters representing decimal numbers
- 3) Interpretation of the sign and significand fields of NaNs
- 4) Conversion between extended precision (3.2) floating-point numbers and decimal strings

2. Definitions

destination: The location for the result of a binary or unary operation. A destination may be either explicitly designated by the user or implicitly supplied by the system (that is, intermediate results in subexpressions or arguments for procedures). Some languages place the results of intermediate calculations in destinations beyond the user's