BRITISH STANDARD

BS ISO/IEC 10967-3:2006

Incorporating corrigendum no. 1

Information technology — Language independent arithmetic —

Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions

ICS 35.060

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI Documents Document Sharing Hub British Standards

Full document details: BS ISO/IEC 10967-3:2006

National foreword

This British Standard is the UK implementation of ISO/IEC 10967-3:2006.

The UK participation in its preparation was entrusted to Technical Committee IST/5, Programming languages, their environments and system software interfaces.

A list of organizations represented on this committee can be obtained on request to its secretary.

NOTE An editorial error has been identified in note 1 of clause 2 of the ISO/IEC text of this standard. The Annex referred to does not exist in this standard. ISO/IEC technical committee JTC1/SC22 has been informed of this error by BSI.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2007

© BSI 2007

ISBN 978 0 580 60212 2

Amendments issued since publication

Amd. No.	Date	Comments
17397 Corrigendum No. 1	28 September 2007	Correction to ISBN

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

INTERNATIONAL STANDARD

ISO/IEC 10967-3

First edition
2006-05-01

Information technology — Language independent arithmetic —

Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions

Technologies de l'information — Arithmétique indépendante des langages —

Partie 3: Arithmétique des nombres complexes entiers et en virgule flottante et fonctions numériques élémentaires complexes

Reference number
ISO/IEC 10967-3:2006(E)

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

Contents

Foreword vii

Introduction viii

1 Scope 1
1.1 Inclusions . . . 1
1.2 Exclusions . . . 2

2 Conformity 3

3 Normative references 4

4 Symbols and definitions 4
4.1 Symbols . . . 4
4.1.1 Sets and intervals . . . 4
4.1.2 Operators and relations . . . 4
4.1.3 Mathematical functions . . . 5
4.1.4 Exceptional values . . . 6
4.1.5 Datatypes and special values . . . 6
4.1.6 Complex value constructors and complex datatype constructors . . . 8
4.2 Definitions of terms . . . 9

5 Specifications for imaginary and complex datatypes and operations 14
5.1 Imaginary and complex integer datatypes and operations . . . 14
5.1.1 The complex integer *result* helper function . . . 15
5.1.2 Imaginary and complex integer operations . . . 15
5.1.2.1 Complex integer comparisons . . . 15
5.1.2.2 Multiplication by the imaginary unit . . . 17
5.1.2.3 The real and imaginary parts of a complex value . . . 17
5.1.2.4 Formation of a complex integer from two real valued integers . . . 18
5.1.2.5 Basic complex integer arithmetic . . . 18
5.1.2.6 Absolute value and signum of integers and imaginary integers . . . 21
5.1.2.7 Divisibility interrogation . . . 21
5.1.2.8 Integer division and remainder extended to imaginary and complex integers . . . 22
5.1.2.9 Maximum and minimum . . . 27
5.2 Imaginary and complex floating point datatypes and operations . . . 28
5.2.1 Maximum error requirements . . . 28
5.2.2 Sign requirements . . . 29
5.2.3 Monotonicity requirements . . . 30
5.2.4 The complex floating point *result* helper functions . . . 30
5.2.5 Basic arithmetic for complex floating point . . . 31
5.2.5.1 Complex floating point comparisons . . . 31
5.2.5.2 Multiplication by the imaginary unit . . . 33
5.2.5.3 The real and imaginary parts of a complex value . . . 34
5.2.5.4 Formation of a complex floating point from two floating point values 34
5.2.5.5 Fundamental complex floating point arithmetic . . . 34

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

5.2.5.6 Absolute value, phase and signum of complex floating point values 38
5.2.5.7 Floor, round, and ceiling . . . 39
5.2.5.8 Maximum and minimum . . . 39
5.2.6 Complex sign, multiplication, and division . . . 40
5.2.6.1 Complex signum . . . 41
5.2.6.2 Complex multiplication . . . 41
5.2.6.3 Complex division . . . 42
5.2.7 Operations for conversion from polar to Cartesian . . . 43
5.3 Elementary transcendental imaginary and complex floating point operations . . . 44
5.3.1 Operations for exponentiations and logarithms . . . 44
5.3.1.1 Exponentiation of imaginary base to integer power . . . 44
5.3.1.2 Natural exponentiation . . . 45
5.3.1.3 Complex exponentiation of argument base . . . 45
5.3.1.4 Complex square root . . . 46
5.3.1.5 Natural logarithm . . . 47
5.3.2 Operations for radian trigonometric elementary functions . . . 49
5.3.2.1 Radian angle normalisation . . . 49
5.3.2.2 Radian sine . . . 49
5.3.2.3 Radian cosine . . . 50
5.3.2.4 Radian tangent . . . 50
5.3.2.5 Radian cotangent . . . 51
5.3.2.6 Radian secant . . . 52
5.3.2.7 Radian cosecant . . . 52
5.3.2.8 Radian arc sine . . . 53
5.3.2.9 Radian arc cosine . . . 54
5.3.2.10 Radian arc tangent . . . 56
5.3.2.11 Radian arc cotangent . . . 57
5.3.2.12 Radian arc secant . . . 58
5.3.2.13 Radian arc cosecant . . . 59
5.3.3 Operations for hyperbolic elementary functions . . . 60
5.3.3.1 Hyperbolic normalisation . . . 61
5.3.3.2 Hyperbolic sine . . . 61
5.3.3.3 Hyperbolic cosine . . . 61
5.3.3.4 Hyperbolic tangent . . . 61
5.3.3.5 Hyperbolic cotangent . . . 62
5.3.3.6 Hyperbolic secant . . . 62
5.3.3.7 Hyperbolic cosecant . . . 62
5.3.3.8 Inverse hyperbolic sine . . . 62
5.3.3.9 Inverse hyperbolic cosine . . . 63
5.3.3.10 Inverse hyperbolic tangent . . . 63
5.3.3.11 Inverse hyperbolic cotangent . . . 64
5.3.3.12 Inverse hyperbolic secant . . . 64
5.3.3.13 Inverse hyperbolic cosecant . . . 65
5.4 Operations for conversion between imaginary and complex numeric datatypes . . . 65
5.4.1 Integer to complex integer conversions . . . 65
5.4.2 Floating point to complex floating point conversions . . . 66
5.5 Support for imaginary and complex numerals . . . 67

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

6 Notification **68**
6.1 Continuation values . . . 68

7 Relationship with language standards **68**

8 Documentation requirements **69**

Annex A (normative) **Partial conformity** **71**
A.1 Maximum error relaxation . . . 71
A.2 Extra accuracy requirements relaxation . . . 71
A.3 Relationships to other operations relaxation . . . 72
A.4 Part 1 and part 2 requirements relaxation . . . 72

Annex B (informative) **Rationale** **73**
B.1 Scope . . . 73
B.1.1 Inclusions . . . 73
B.1.2 Exclusions . . . 73
B.2 Conformity . . . 73
B.3 Normative references . . . 74
B.4 Symbols and definitions . . . 74
B.4.1 Symbols . . . 74
B.4.1.1 Sets and intervals . . . 74
B.4.1.2 Operators and relations . . . 74
B.4.1.3 Mathematical functions . . . 74
B.4.1.4 Exceptional values . . . 75
B.4.1.5 Datatypes and special values . . . 75
B.4.1.6 Complex value constructors and complex datatype constructors . 75
B.4.2 Definitions of terms . . . 75
B.5 Specifications for the imaginary and complex datatypes and operations . . . 76
B.5.1 Imaginary and complex integer datatypes and operations . . . 76
B.5.2 Imaginary and complex floating point datatypes and operations . . . 76
B.5.2.1 Maximum error requirements . . . 76
B.5.2.2 Sign requirements . . . 76
B.5.2.3 Maximum error requirements . . . 76
B.5.2.4 Basic arithmetic for complex floating point . . . 77
B.5.3 Elementary transcendental imaginary and complex floating point operations 78
B.5.3.1 Operations for exponentiations and logarithms . . . 78
B.5.3.2 Operations for radian trigonometric elementary functions . . . 78
B.5.3.2.1 Radian angle normalisation . . . 78
B.5.3.2.2 Radian sine . . . 79
B.5.3.2.3 Radian cosine . . . 79
B.5.3.2.4 Radian tangent . . . 79
B.5.3.2.5 Radian cotangent . . . 80
B.5.3.2.6 Radian secant . . . 80
B.5.3.2.7 Radian cosecant . . . 80
B.5.3.2.8 Radian arc sine . . . 80
B.5.3.2.9 Radian arc cosine . . . 81
B.5.3.2.10 Radian arc tangent . . . 81
B.5.3.2.11 Radian arc cotangent . . . 81

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

B.5.3.2.12 Radian arc secant 81
B.5.3.2.13 Radian arc cosecant 81
B.5.3.3 Operations for hyperbolic elementary functions 82
B.5.3.3.1 Hyperbolic normalisation 82
B.5.3.3.2 Hyperbolic sine 82
B.5.3.3.3 Hyperbolic cosine 82
B.5.3.3.4 Hyperbolic tangent 82
B.5.3.3.5 Hyperbolic cotangent 83
B.5.3.3.6 Hyperbolic secant 83
B.5.3.3.7 Hyperbolic cosecant 83
B.5.3.3.8 Inverse hyperbolic sine 83
B.5.3.3.9 Inverse hyperbolic cosine 83
B.5.3.3.10 Inverse hyperbolic tangent 84
B.5.3.3.11 Inverse hyperbolic cotangent 84
B.5.3.3.12 Inverse hyperbolic secant 84
B.5.3.3.13 Inverse hyperbolic cosecant 84
B.5.4 Operations for conversion between imaginary and complex numeric datatypes 84
B.5.5 Support for imaginary and complex numerals 84
B.6 Notification 84
B.6.1 Continuation values 84
B.7 Relationship with language standards 84
B.8 Documentation requirements 84

Annex C (informative) **Example bindings for specific languages** **85**
C.1 Ada 86
C.2 C 96
C.3 C++ 105
C.4 Fortran 113
C.5 Common Lisp 122

Annex D (informative) **Bibliography** **133**

Annex E (informative) **Cross reference** **135**

Annex F (informative) **Possible changes to part 2** **147**

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

IISO/IEC 10967-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces.*

ISO/IEC 10967 consists of the following parts, under the general title *Information technology — Language independent arithmetic*:

— *Part 1: Integer and floating point arithmetic*

— *Part 2: Elementary numerical functions*

— *Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions*

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

Introduction

The aims

Portability is a key issue for scientific and numerical software in today's heterogeneous computing environment. Such software may be required to run on systems ranging from personal computers to high performance pipelined vector processors and massively parallel systems, and the source code may be ported between several programming languages.

Part 1 of ISO/IEC 10967 specifies the basic properties of integer and floating point types that can be relied upon in writing portable software.

Part 2 of ISO/IEC 10967 specifies a number of additional operations for integer and floating point types, in particular specifications for numerical approximations to elementary functions on reals.

The content

The content of this document is based on part 1 and part 2, and extends part 1's and part 2's specifications to also cover operations approximating imaginary-integer and complex-integer arithmetic, imaginary-real and complex-real arithmetic, as well as imaginary-real and complex-real elementary functions.

The numerical functions covered by this document are computer approximations to mathematical functions of one or more imaginary or complex arguments. Accuracy versus performance requirements often vary with the application at hand. This is recognised by recommending that implementors support more than one library of these numerical functions. Various documentation and (program available) parameters requirements are specified to assist programmers in the selection of the library best suited to the application at hand.

The benefits

Adoption and proper use of this document can lead to the following benefits.

For programming language standards it will be possible to define their arithmetic semantics more precisely without preventing the efficient implementation of the language on a wide range of machine architectures.

Programmers of numeric software will be able to assess the portability of their programs in advance. Programmers will be able to trade off program design requirements for portability in the resulting program. Programs will be able to determine (at run time) the crucial numeric properties of the implementation. They will be able to reject unsuitable implementations, and (possibly) to correctly characterize the accuracy of their own results. Programs will be able to detect (and possibly correct for) exceptions in arithmetic processing.

Procurers of numerical programs will find it easier to determine whether a (properly documented) application program is likely to execute satisfactorily on the platform used. This can be done by comparing the documented requirements of the program against the documented properties of the platform.

Finally, end users of numeric application packages will be able to rely on the correct execution of those packages. That is, for correctly programmed algorithms, the results are reliable if and only if there is no notification.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

Information technology — Language independent arithmetic —

Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions

1 Scope

This part of ISO/IEC 10967 specifies the properties of numerical approximations for complex arithmetic operations and many of the complex elementary numerical functions available in a variety of programming languages in common use for mathematical and numerical applications.

An implementor may choose any combination of hardware and software support to meet the specifications of this document. It is the computing environment, as seen by the programmer/user, that does or does not conform to the specifications.

The term *implementation* (of this part) denotes the total computing environment pertinent to this part, including hardware, language processors, subroutine libraries, exception handling facilities, other software, and documentation.

1.1 Inclusions

The specifications of part 1 and part 2 are included by reference in this part.

This document provides specifications for properties of complex and imaginary integer datatypes and floating point datatypes, basic operations on values of these datatypes as well as for some numerical functions for which operand or result values are of imaginary or complex integer datatypes or imaginary or complex floating point datatypes constructed from integer and floating point datatypes satisfying the requirements of part 1 (ISO/IEC 10967-1). Boundaries for the occurrence of exceptions and the maximum error allowed are prescribed for each specified operation. Also the result produced by giving a special value operand, such as an infinity, or a NaN (not-a-number), is prescribed for each specified floating point operation.

This document provides specifications for:

a) Basic imaginary integer and complex integer operations.

b) Non-transcendental imaginary floating point and Cartesian complex floating point operations.

c) Exponentiation, logarithm, radian trigonometric, and hyperbolic operations for imaginary floating point and Cartesian complex floating point.

This document also provides specifications for:

d) The results produced by an included floating point operation when one or more operand values include IEC 60559 special values.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

e) Program-visible parameters that characterise certain aspects of the operations.

1.2 Exclusions

This document provides no specifications for:

a) Datatypes and operations for polar complex floating point. This part neither requires nor excludes the presence of such polar complex datatypes and operations.

b) Numerical functions whose operands are of more than one datatype, except certain imaginary/complex combinations. This part neither requires nor excludes the presence of such "mixed operand" operations.

c) A complex interval datatype, or the operations on such datatypes. This part neither requires nor excludes such datatypes or operations.

d) A complex fixed point datatype, or the operations on such datatypes. This part neither requires nor excludes such datatypes or operations.

e) A complex rational datatype, or the operations on such datatypes. This part neither requires nor excludes such datatypes or operations.

f) Matrix, statistical, or symbolic operations (on suitable datatypes). This part neither requires nor excludes such operations or datatypes.

g) The properties of complex arithmetic datatypes that are not related to the numerical process, such as the representation of values on physical media.

h) The properties of integer and floating point datatypes that properly belong in programming language standards or other specifications. Examples include:

   1) the syntax of numerals and expressions in the programming language,
   2) the syntax used for parsed (input) or generated (output) character string forms for numerals by any specific programming language or library,
   3) the precedence of operators in the programming language,
   4) the rules for assignment, parameter passing, and returning value,
   5) the presence or absence of automatic datatype coercions,
   6) the consequences of applying an operation to values of improper datatype, or to uninitialised data.

Furthermore, this part does not provide specifications for how the operations should be implemented or which algorithms are to be used for the various operations.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

2 Conformity

It is expected that the provisions of this document will be incorporated by reference and further defined in other International Standards; specifically in programming language standards and in binding standards.

A binding standard specifies the correspondence between one or more of the arithmetic datatypes, parameters, and operations specified in this document and the concrete language syntax of some programming language. More generally, a binding standard specifies the correspondence between certain datatypes, parameters, and operations and the elements of some arbitrary computing entity. A language standard that explicitly provides such binding information can serve as a binding standard.

When a binding standard for a language exists, an implementation shall be said to conform to this document if and only if it conforms to the binding standard. In case of conflict between a binding standard and this document, the specifications of the binding standard take precedence.

When a binding standard requires only a subset of the imaginary or complex integer or imaginary or complex floating point datatypes specified in this document, an implementation remains free to conform to this document with respect to other datatypes independently of that binding standard.

When a binding standard requires only a subset of the operations specified in this document, an implementation remains free to conform to this document with respect to other operations, independently of that binding standard.

When no binding standard between a language and some datatypes or operations specified in this document exists, an implementation conforms to this document if and only if it provides one or more datatypes and one or more operations that together satisfy all the requirements of clauses 5 through 8 that are relevant to those datatypes and operations. The implementation shall then document the binding.

Conformity to this document is always with respect to a specified set of datatypes and set of operations. Conformity to this document implies conformity to part 1 and part 2 of ISO/IEC 10967 for the integer and floating point datatypes and operations used. Under certain circumstances, conformity to IEC 60559 is implied by conformity to part 1 of ISO/IEC 10967.

An implementation is free to provide arithmetic datatypes and arithmetic operations that do not conform to this document, or that are beyond the scope of this document. The implementation shall not claim or imply conformity to this document for such datatypes or operations.

An implementation is permitted to have modes of operation that do not conform to this document. A conforming implementation shall specify how to select the modes of operation that ensure conformity.

NOTES

1 Language bindings are essential. Clause 8 requires an implementation to document a binding if no binding standard exists. See Annex **??** for an example of a conformity statement, and Annex C for suggested language bindings.

2 A complete binding for this document will include (explicitly or by reference) a binding for part 2 and part 1 as well, which in turn may include (explicitly or by reference) a binding for IEC 60559 as well.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

3 This document does not require a particular set of operations to be provided. It is not possible to conform to this document without specifying to which datatypes and set of operations (and modes of operation) conformity is claimed.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems.*

ISO/IEC 10967-1, *Information technology – Language independent arithmetic – Part 1: Integer and floating point arithmetic.*

ISO/IEC 10967-2, *Information technology – Language independent arithmetic – Part 2: Elementary numerical functions.*

4 Symbols and definitions

4.1 Symbols

4.1.1 Sets and intervals

In this document, $\mathcal{Z}$ denotes the set of mathematical integers, $\mathcal{G}$ denotes the set of complex integers. $\mathcal{R}$ denotes the set of classical real numbers, and $\mathcal{C}$ denotes the set of complex numbers over $\mathcal{R}$. Note that $\mathcal{Z} \subset \mathcal{R} \subset \mathcal{C}$, and $\mathcal{Z} \subset \mathcal{G} \subset \mathcal{C}$.

The conventional notation for set definition and manipulation is used.

The following notation for intervals is used:

$[x, z]$ designates the interval $\{y \in \mathcal{R} \mid x \leqslant y \leqslant z\}$,
$]x, z]$ designates the interval $\{y \in \mathcal{R} \mid x < y \leqslant z\}$,
$[x, z[$ designates the interval $\{y \in \mathcal{R} \mid x \leqslant y < z\}$, and
$]x, z[$ designates the interval $\{y \in \mathcal{R} \mid x < y < z\}$.

NOTE – The notation using a round bracket for an open end of an interval is not used, for the risk of confusion with the notation for pairs.

4.1.2 Operators and relations

All prefix and infix operators have their conventional (exact) mathematical meaning. The conventional notation for set definition and manipulation is also used. In particular:

$\Rightarrow$ and $\Leftrightarrow$ for logical implication and equivalence
$+$, $-$, $/$, $|x|$, conj, $\lfloor x \rfloor$, $\lceil x \rceil$, and round(x) on complex values
$\cdot$ for multiplication on complex values
$<$, $\leqslant$, $\geqslant$, and $>$ between real values
$=$ and $\neq$ between real, complex, as well as special values

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

$\cup$, $\cap$, $\in$, $\notin$, $\subset$, $\subseteq$, $\not\subseteq$, $\neq$, and $=$ with sets
$\times$ for the Cartesian product of sets
$\rightarrow$ for a mapping between sets
$|$ for the divides relation between complex integer values (in $\mathcal{G}$)
$\tilde{\imath}$ as the imaginary unit ($\tilde{\imath}^2 = -1$)
$\mathfrak{Re}$ to extract the real part of a complex value (in $\mathcal{C}$)
$\mathfrak{Im}$ to extract the imaginary part of a complex value (in $\mathcal{C}$)

NOTE 1 – $\approx$ is used informally, in notes and the rationale.

For $x \in \mathcal{C}$, the notation $\lfloor x \rfloor$ designates the component-wise largest complex integer where each part is not greater than the corresponding part of x:

$$\lfloor x \rfloor \in \mathcal{G} \text{ and } \mathfrak{Re}(x) - 1 < \mathfrak{Re}(\lfloor x \rfloor) \leqslant \mathfrak{Re}(x) \text{ and } \mathfrak{Im}(x) - 1 < \mathfrak{Im}(\lfloor x \rfloor) \leqslant \mathfrak{Im}(x)$$

the notation $\lceil x \rceil$ designates the component-wise smallest complex integer where each part is not less than the corresponding part of x:

$$\lceil x \rceil \in \mathcal{G} \text{ and } \mathfrak{Re}(x) \leqslant \mathfrak{Re}(\lceil x \rceil) < \mathfrak{Re}(x) + 1 \text{ and } \mathfrak{Im}(x) \leqslant \mathfrak{Im}(\lceil x \rceil) < \mathfrak{Im}(x) + 1$$

and the notation round(x) designates the complex integer closest to x:

$$\text{round}(x) \in \mathcal{G} \text{ and}$$
$$\mathfrak{Re}(x) - 0.5 \leqslant \mathfrak{Re}(\text{round}(x)) \leqslant \mathfrak{Re}(x) + 0.5 \text{ and } \mathfrak{Im}(x) - 0.5 \leqslant \mathfrak{Im}(\text{round}(x)) \leqslant \mathfrak{Im}(x) + 0.5$$

where in case $\mathfrak{Re}(x)$ or $\mathfrak{Im}(x)$ is exactly half-way between two integers, the even integer is the result component.

The *divides* relation ($|$) on complex integers tests whether a complex integer i divides a complex integer j exactly:

$$i|j \;\Leftrightarrow\; (i \neq 0 \text{ and } i \cdot n = j \text{ for some } n \in \mathcal{G})$$

NOTE 2 – $i|j$ is true exactly when j/i is defined and $j/i \in \mathcal{G}$).

4.1.3 Mathematical functions

This document specifies properties for a number of operations numerically approximating some of the elementary functions. The following ideal mathematical functions are defined in chapter 4 of the *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* [11] (e is the Napierian base):

e^x, x^y, $\sqrt{x}$, $|x|$, ln, $\log_b$,
sin, cos, tan, cot, sec, csc, arcsin, arccos, arctan, arccot, arcsec, arccsc,
sinh, cosh, tanh, coth, sech, csch, arcsinh, arccosh, arctanh, arccoth, arcsech, arccsch.

Many of the inverses above are multi-valued. The selection of which value to return, the principal value, so as to make the inverses into functions, is done in the conventional way. E.g., $\sqrt{x} \in [0, \infty[$ when $x \in [0, \infty[$.

Part 2 defines the mathematical *arc* function, which is used in this part to define the mathematical complex *signum* function.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

4.1.4 Exceptional values

ISO/IEC 10967 uses the following five exceptional values:

underflow: the result has an absolute value that is smaller than the smallest positive normalised value, and the result may be inexact. This notification need not be given if the result is exact. In particular, if the result is zero, or exact and IEC 60559 is conformed to and trapping is not enabled.

overflow: the result, after rounding, has a magnitude larger than the maximum value representable in the result datatype.

infinitary: the operation result is (exactly) infinite (in its real or its imaginary part), while all the arguments (their real and imaginary parts) are finite. This is to say that the corresponding mathematical function has a pole at the finite argument point.

invalid: the operation is undefined, while not infinitary, for the given arguments.

absolute_precision_underflow: indicate that the argument is such that the density of representable argument values is too small in the neighbourhood of the given argument value for a numeric result to be considered appropriate to return. Used for operations that approximate trigonometric functions (part 2 and part 3), and hyperbolic and exponentiation functions (part 3).

The exceptional value **inexact** is not specified in ISO/IEC 10967, but IEC 60559 conforming implementations will provide it. It should then be used also for operations approximating transcendental functions, when the returned result may be approximate. This part of ISO/IEC 10967 does not specify when it is appropriate to return this exceptional value, but does specify an appropriate continuation value (see Definition 4.2.4). Thus, v is specified by ISO/IEC 10967 when v or **inexact**(v) should be returned by implementations that are based on IEC 60559, and ISO/IEC 10967 specifies **underflow**(v) when **underflow**(v) or **inexact**(**underflow**(v)) should be returned by implementations that are based on IEC 60559. Ideally, **underflow** should imply **inexact**.

For the exceptional values, a continuation value may be given in this part in parenthesis after the exceptional value.

4.1.5 Datatypes and special values

The datatype **Boolean** consists of the two values **true** and **false**.

NOTE 1 – Mathematical relations are true or false (or undefined, if an operand is undefined), which are not values. In contrast, **true** and **false** are values in **Boolean**.

Square brackets are used to write finite sequences of values. [] is the sequence containing no values. $[s]$, is the sequence of one value, s. $[s_1, s_2]$, is the sequence of two values, s_1 and then s_2, etc. The colon operator is used to prepend a value to a sequence: $x : [x_1, ..., x_n] = [x, x_1, ..., x_n]$. $[S]$, where S is a set, denotes the set of finite sequences, where each value in a sequence is in S.

NOTE 2 – It is always clear from context, in the text of this part, if $[X]$ is a sequence of one element, or the set of sequences with values from X. It is also clear from context if $[x_1, x_2]$ is a sequence of two values or an interval.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

Integer datatypes and floating point datatypes are defined in part 1. Let I be the non-special value set for an integer datatype conforming to part 1. Let F be the non-special value set for a floating point datatype conforming to part 1 and part 2. The following symbols used in this part are defined in part 1 or part 2:

Exceptional values:
underflow, **overflow**, **infinitary**, **invalid**, and **absolute_precision_underflow**.
Integer helper function:
$result_I$.
Integer operations:
eq_I, neq_I, lss_I, leq_I, gtr_I, geq_I, neg_I, add_I, sub_I, mul_I, abs_I, max_I, min_I, max_seq_I, min_seq_I, $divides_I$, $quot_I$, mod_I, $ratio_I$, $residue_I$, $group_I$, and pad_I.
Integer conversion operation:
$convert_{I \to I'}$.
Floating point parameters:
r_F, p_F, $emin_F$, $emax_F$, $denorm_F$, and iec_559_F.
Derived floating point constants:
$fmax_F$, $fmin_F$, $fminN_F$, $fminD_F$, and $epsilon_F$.
Floating point value sets related to F:
F^*, F_D, F_N, G_F, $F^{2\cdot\pi}$, and F^u (for a given u).
Floating point helper functions:
up_F, $down_F$, $nearest_F$, $result_F$, $result^*_F$, no_result_F, and $no_result2_F$.
Floating point operations from part 1:
eq_F, neq_F, lss_F, leq_F, gtr_F, geq_F, neg_F, add_F, sub_F, mul_F, div_F, abs_F, max_F, $mmax_F$, min_F, $mmin_F$, max_seq_F, $mmax_seq_F$, min_seq_F, $mmin_seq_F$, $floor_F$, $ceiling_F$, and $rounding_F$.
Floating point operations from part 2:
$sqrt_F$, $hypot_F$, $power_{F,I}$, exp_F, $power_F$, ln_F, $logbase_F$, rad_F, sin_F, cos_F, tan_F, cot_F, sec_F, csc_F, $arcsin_F$, $arccos_F$, $arctan_F$, $arccot_F$, $arcsec_F$, $arccsc_F$, arc_F, $arcu_F$, $sinu_F$, $cosu_F$, $sinh_F$, $cosh_F$, $tanh_F$, $coth_F$, $sech_F$, $csch_F$, $arcsinh_F$, $arccosh_F$, $arctanh_F$, $arccoth_F$, $arcsech_F$, and $arccsch_F$.
Angular parameters and maximum error parameters from part 2:
$big_angle_r_F$, $big_angle_u_F$, $max_error_tan_F$, and $max_error_tanu_F$.
Floating point conversion operations:
$convert_{F \to F'}$, $convert_{F \to D}$, and $convert_{D \to F}$.
Approximation helper functions from part 2:
exp^*_F, $power^*_F$, ln^*_F, $logbase^*_F$, sin^*_F, cos^*_F, tan^*_F, cot^*_F, sec^*_F, csc^*_F, $arcsin^*_F$, $arccos^*_F$, $arctan^*_F$, $arccot^*_F$, $arcsec^*_F$, $arccsc^*_F$, $sinh^*_F$, $cosh^*_F$, $tanh^*_F$, $coth^*_F$, $sech^*_F$, $csch^*_F$, $arcsinh^*_F$, $arccosh^*_F$, $arctanh^*_F$, $arccoth^*_F$, $arcsech^*_F$, and $arccsch^*_F$.

Floating point datatypes that conform to part 1 shall, for use with this part, like for part 2, have a value for the parameter p_F such that $p_F \geqslant 2 \cdot \max\{1, \log_{r_F}(2 \cdot \pi)\}$, and have a value for the parameter $emin_F$ such that $emin_F \leqslant -p_F - 1$.

NOTES

3 This implies that $fminN_F < 0.5 \cdot epsilon_F / r_F$ in this part, rather than just $fminN_F \leqslant epsilon_F$.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

4 These extra requirements, which do not limit the use of any existing floating point datatype, are made so that angles in radians are not too degenerate within the first two cycles, plus and minus, when represented in F.

5 F should also be such that $p_F \geqslant 2 + \log_{r_F}(1000)$, to allow for a not too coarse angle resolution anywhere in the interval $[-big_angle_r_F, big_angle_r_F]$ with the default value for $big_angle_r_F$. See Clause 5.3.9 of part 2.

The following symbols represent special values defined in IEC 60559 and are used in this part:

$\mathbf{-0}$, $\mathbf{+\infty}$, $\mathbf{-\infty}$, **qNaN**, and **sNaN**.

These floating point values are not part of the set F, but if iec_559_F has the value **true**, these values are included in the floating point datatype corresponding to F.

NOTE 6 – This part uses the above five special values for compatibility with IEC 60559. In particular, the symbol $\mathbf{-0}$ (in bold) is not the application of (mathematical) unary − to the value 0, and is a value logically distinct from 0.

The specifications cover the results to be returned by an operation if given one or more of the IEC 60559 special values $\mathbf{-0}$, $\mathbf{+\infty}$, $\mathbf{-\infty}$, or NaNs as input values. These specifications apply only to systems which provide and support these special values. If an implementation is not capable of representing a $\mathbf{-0}$ result or continuation value, 0 shall be used as the actual result or continuation value. If an implementation is not capable of representing a prescribed result or continuation value of the IEC 60559 special values $\mathbf{+\infty}$, $\mathbf{-\infty}$, or **qNaN**, the actual result or continuation value is binding or implementation defined.

If and only if an implementation is not capable of representing $\mathbf{-0}$:

a) a 0 as the imaginary part of a complex argument (in $\mathrm{c}(F)$, see Clause 4.1.6) shall be interpreted as if it was $\mathbf{-0}$ if and only if the real part of that complex argument is greater than or equal to zero, and

b) a 0 as the real part of a complex argument (in $\mathrm{c}(F)$, see Clause 4.1.6) shall be interpreted as if it was $\mathbf{-0}$ if and only if the imaginary part of the complex argument is less than zero.

NOTES

7 Reinterpreting 0 as $\mathbf{-0}$ as required above is needed to follow the sign rules for inverse trigonometric and inverse hyperbolic operations, as well as the exact relations between trigonometric and hyperbolic operations also for argument parts (real and imaginary) that have a zero as value.

8 The rule above is sometimes referred to as *continuous when approaching an axis in a counterclockwise path*. This fits both with Common Lisp and C requirements when zeroes don't have a distinguishable sign.

9 For consistency, this rule also has implications for the operations that implicitly or explicitly extract an implicit real or implicit imaginary part (see for example the specifications for the $re_{\mathrm{i}(F)}$ and im_F operations in Clause 5.2.5).

4.1.6 Complex value constructors and complex datatype constructors

Let X be a set containing values in $\mathcal{R}$, and possibly also containing special values (such as IEC 60559 special values).

$\mathrm{i}(X)$ is a subset of values in an imaginary datatype, constructed from the datatype corresponding to X. $\hat{\mathbf{i}}\cdot$ is a prefix constructor that takes one parameter.

$$\mathrm{i}(X) = \{\hat{\mathbf{i}}\cdot y \mid y \in X\}$$

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

$\mathrm{c}(X)$ is a subset of values in a complex datatype, constructed from the datatype corresponding to X. $\mathbf{+\hat{\imath}\cdot}$ is an infix constructor that takes two parameters.

$$\mathrm{c}(X) = \{x \mathbf{+\hat{\imath}\cdot} y \mid x, y \in X\}$$

NOTES

1 While $\mathbf{\hat{\imath}\cdot}$ and $\mathbf{+\hat{\imath}\cdot}$ (note that they are written in bold) have an appearance of being the imaginary unit together with the plus and times operators, that is not the case. For instance, $\mathbf{\hat{\imath}\cdot} 2$ is an element of $\mathrm{i}(X)$ (if $2 \in X$), but not of $\mathcal{G}$ or $\mathcal{C}$. $\tilde{\imath} \cdot 2$, on the other hand, is an expression that denotes an element of $\mathcal{G}$ (and $\mathcal{C}$), but neither of $\mathrm{i}(X)$ nor $\mathrm{c}(X)$. Further, e.g., $4 + \tilde{\imath} \cdot 0 = 4$, but $4 \mathbf{+\hat{\imath}\cdot} 0 \neq 4$.

2 A constructor that takes one argument is a one-tuple tag. A constructor that takes two arguments is a two-tuple (pair) tag. The arguments are part of the resulting value.

3 The tuple tags need not be explicitly represented in implementations. But if represented, there should be different tags for different argument types (which is not needed in this text).

Some of the helper function signatures use $\mathcal{C}_F$, where

$$\mathcal{C}_F = \{x + \tilde{\imath} \cdot y \mid x, y \in F\}$$

where $F \subset \mathcal{R}$.

4.2 Definitions of terms

For the purposes of this document, the following terms and definitions apply.

4.2.1
accuracy

The closeness between the true mathematical result and a computed result.

4.2.2
arithmetic datatype

A datatype whose non-special values are members of $\mathcal{Z}$, $\mathrm{i}(\mathcal{Z})$, $\mathrm{c}(\mathcal{Z})$, $\mathcal{R}$, $\mathrm{i}(\mathcal{R})$, or $\mathrm{c}(\mathcal{R})$.

NOTE 1 – $\mathrm{i}(\mathcal{Z})$ corresponds to imaginary integer values in $\mathcal{G}$. $\mathrm{c}(\mathcal{Z})$ corresponds to complex integer values in $\mathcal{G}$. $\mathrm{i}(\mathcal{R})$ corresponds to imaginary values in $\mathcal{C}$. $\mathrm{c}(\mathcal{R})$ corresponds to complex values in $\mathcal{C}$.

4.2.3
binding

Documentation that specifies which syntax (most often, operations or functions) of a programming language, or programming library, that is used for an operation specified in this part.

4.2.4
continuation value

A computational value used as the result of an arithmetic operation when an exception occurs. Continuation values are intended to be used in subsequent arithmetic processing. A continuation value can be a (in the datatype representable) value in $\mathcal{Z}$, $\mathrm{i}(\mathcal{Z})$, $\mathrm{c}(\mathcal{Z})$, $\mathcal{R}$, $\mathrm{i}(\mathcal{R})$, $\mathrm{c}(\mathcal{R})$, or a value where one or both parts of the value is an IEC 60559 special value (e.g. $\mathbf{-0}$, $\mathbf{\hat{\imath}\cdot}\,(\mathbf{-\infty})$), or $\mathbf{+\infty \mathbf{+\hat{\imath}\cdot} qNaN}$). (Contrast with *exceptional value*. See Clause 6.1.2 of part 1.)

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

4.2.5
denormalisation loss
A larger than normal rounding error caused by the fact that subnormal values have less than full precision. (See Clause 5.2.5 of part 1 for a full definition.)

4.2.6
error
⟨in computed value⟩ The difference between a computed value and the mathematically correct value. Used in phrases like "rounding error" or "error bound".

4.2.7
error
⟨computation gone awry⟩ A synonym for *exception* in phrases like "error message" or "error output". Error and exception are not synonyms in any other contexts.

4.2.8
exception
The inability of an operation to return a suitable finite numeric result from finite arguments. This might arise because no such finite result exists mathematically, or because the mathematical result cannot be represented with sufficient accuracy.

4.2.9
exceptional value
A non-numeric value produced by an arithmetic operation to indicate the occurrence of an exception. Exceptional values are not used in subsequent arithmetic processing. (See Clause 5 of part 1.)

NOTES

2 Exceptional values are used as part of the defining formalism only. With respect to this part, they do not represent values of any of the datatypes described. There is no requirement that they be represented or stored in the computing system.

3 Exceptional values are not to be confused with the NaNs and infinities defined in IEC 60559. Contrast this definition with that of *continuation value* above.

4.2.10
helper function
A function used solely to aid in the expression of a requirement. Helper functions are not visible to the programmer, and are not required to be part of an implementation.

4.2.11
implementation (of this part)
The total arithmetic environment presented to a programmer, including hardware, language processors, exception handling facilities, subroutine libraries, other software, and all pertinent documentation.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

4.2.12
literal
A syntactic entity denoting a constant value without having proper sub-entities that are expressions.

4.2.13
monotonic approximation
An approximation helper function $h : ... \times S \times ... \rightarrow \mathcal{R}$, where the other arguments are kept constant, and where $S \subseteq \mathcal{R}$, is a monotonic approximation of a predetermined mathematical function $f : \mathcal{R} \rightarrow \mathcal{R}$ if, for every $a \in S$ and $b \in S$, where $a < b$,

a) f is monotonic non-decreasing on $[a, b]$ implies $h(..., a, ...) \leqslant h(..., b, ...)$,

b) f is monotonic non-increasing on $[a, b]$ implies $h(..., a, ...) \geqslant h(..., b, ...)$.

4.2.14
monotonic non-decreasing
A function $f : \mathcal{R} \rightarrow \mathcal{R}$ is monotonic non-decreasing on a real interval $[a, b]$ if for every x and y such that $a \leqslant x \leqslant y \leqslant b$, $f(x)$ and $f(y)$ are well-defined and $f(x) \leqslant f(y)$.

4.2.15
monotonic non-increasing
A function $f : \mathcal{R} \rightarrow \mathcal{R}$ is monotonic non-increasing on a real interval $[a, b]$ if for every x and y such that $a \leqslant x \leqslant y \leqslant b$, $f(x)$ and $f(y)$ are well-defined and $f(x) \geqslant f(y)$.

4.2.16
normalised
The non-zero values of a floating point type F that provide the full precision allowed by that type. (See F_N in Clause 5.2 of part 1 for a full definition.)

4.2.17
notification
The process by which a program (or that program's end user) is informed that an arithmetic exception has occurred. For example, dividing 2 by 0 results in a notification. (See Clause 6 of part 1

4.2.18
numeral
A numeric literal. It may denote a value in $\mathcal{Z}$, i($\mathcal{Z}$), c($\mathcal{Z}$), $\mathcal{R}$, i($\mathcal{R}$), or c($\mathcal{R}$), a value which is $\mathbf{-0}$, an infinity, or a NaN.

4.2.19
numerical function
A computer routine or other mechanism for the approximate evaluation of a mathematical function.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

4.2.20
operation
A function directly available to the programmer, as opposed to helper functions or theoretical mathematical functions.

4.2.21
pole
A mathematical function f has a pole at x_0 if x_0 is finite, f is defined, finite, monotone, and continuous in at least part of the neighbourhood of x_0, and the imaginary or real part of $\lim_{x \to x_0} f(x)$ is infinite via at least one path to x_0.

4.2.22
precision
The number of digits in the fraction of a floating point number. (See Clause 5.2 of part 1.)

4.2.23
rounding
The act of computing a representable final result for an operation that is close to the exact (but unrepresentable in the result datatype) result for that operation. Note that a suitable representable result may not exist (see Clause 5.2.6 of part 1). (See also Annex A.5.2.6 of part 1 for some examples.)

4.2.24
rounding function
Any function $rnd : \mathcal{R} \to X$ (where X is a given discrete and unlimited subset of $\mathcal{R}$) that maps each element of X to itself, and is monotonic non-decreasing. Formally, if x and y are in $\mathcal{R}$,

$$x \in X \Rightarrow rnd(x) = x$$
$$x < y \Rightarrow rnd(x) \leqslant rnd(y)$$

Note that if $u \in \mathcal{R}$ is between two adjacent values in X, $rnd(u)$ selects one of those adjacent values.

4.2.25
round to nearest
The property of a rounding function rnd that when $u \in \mathcal{R}$ is between two adjacent values in X, $rnd(u)$ selects the one nearest u. If the adjacent values are equidistant from u, either may be chosen deterministically.

4.2.26
round toward minus infinity
The property of a rounding function rnd that when $u \in \mathcal{R}$ is between two adjacent values in X, $rnd(u)$ selects the one less than u.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

4.2.27
round toward plus infinity
The property of a rounding function rnd that when $u \in \mathcal{R}$ is between two adjacent values in X, $rnd(u)$ selects the one greater than u.

4.2.28
shall
A verbal form used to indicate requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted. (Quoted from the directives [1].)

4.2.29
should
A verbal form used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that (in the negative form) a certain possibility is deprecated but not prohibited. (Quoted from the directives [1].)

4.2.30
signature (of a function or operation)
A summary of information about an operation or function. A signature includes the function or operation name; a subset of allowed argument values to the function or operation; and a superset of results from the function or operation (including exceptional values if any), if the argument is in the subset of argument values given in the signature.

The signature

$$add_I : I \times I \rightarrow I \cup \{\textbf{overflow}\}$$

states that the operation named add_I shall accept any pair of I values as input, and (when given such input) shall return either a single I value as its output or the exceptional value **overflow**.

A signature for an operation or function does not forbid the operation from accepting a wider range of arguments, nor does it guarantee that every value in the result range will actually be returned for some input. An operation given an argument outside the stipulated argument domain may produce a result outside the stipulated result range.

4.2.31
subnormal
denormal (obsolete)
The values of a floating point datatype F, as well as $\mathbf{-0}$, whose absolute values are strictly less than the smallest positive normal value in F.

NOTE 4 – Compare the definition of F_{D} in Clause 5.2 of part 1. In the first edition (1994) of part 1 and in IEC 60559:1989 this concept, then excepting the zero values, was called denormal.

4.2.32
ulp
The value of one "unit in the last place" of a floating point number. This value depends on the exponent, the radix, and the precision used in representing the number. Thus, the ulp of a

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

normalised value x (in F), with exponent t, precision p, and radix r, is r^{t-p}, and the ulp of a subnormal value is $fminD_F$. (See Clause 5.2 of part 1.)

5 Specifications for imaginary and complex datatypes and operations

This clause specifies imaginary and complex integer datatypes, imaginary and complex floating point datatypes and a number of helper functions and operations for imaginary and complex integer as well as imaginary and complex floating point datatypes.

Each operation is given a signature and is further specified by a number of cases. These cases may refer to other operations (specified in this document, in part 1, or in part 2), to mathematical functions (textbook elementary functions, or functions defined in ISO/IEC 10967), and to helper functions (specified in this document, in part 1, or in part 2). They also use special abstract values ($\mathbf{-\infty}$, $\mathbf{+\infty}$, $\mathbf{-0}$, **qNaN**, **sNaN**). For each datatype, two of these abstract values, **qNaN** and **sNaN**, may represent several actual values each. Finally, the specifications may refer to exceptional values.

The signatures in the specifications in this clause specify only all non-special values as input values, and indicate as output values a superset of all non-special, special, and exceptional values that may result from these (non-special) input values. Exceptional and special values that can never result from non-special input values are not included in the signatures given. Also, signatures that, for example, include IEC 60559 special values as arguments are not given in the specifications below. This does not exclude such signatures from being valid for these operations.

NOTE – For instance, the realpart operation on complex floating point is given with the following signature:

$$re_{\mathrm{c}(F)} : \mathrm{c}(F) \rightarrow F$$

But the following signature is also valid, and takes some special values into account

$$re_{\mathrm{c}(F)} : \mathrm{c}(F \cup \{\mathbf{-\infty}, \mathbf{-0}, \mathbf{+\infty}\}) \rightarrow F \cup \{\mathbf{-\infty}, \mathbf{-0}, \mathbf{+\infty}\}$$

The following signature is also valid

$$re_{\mathrm{c}(F)} : \mathrm{c}(F \cup \{\mathbf{-\infty}, \mathbf{-0}, \mathbf{+\infty}, \mathbf{qNaN}, \mathbf{sNaN}\}) \rightarrow F \cup \{\mathbf{-\infty}, \mathbf{-0}, \mathbf{+\infty}, \mathbf{qNaN}, \mathbf{invalid}\}$$

5.1 Imaginary and complex integer datatypes and operations

Clause 5.1 of part 1 and clause 5.1 of part 2 specify integer datatypes and a number of operations on values of an integer datatype. In this clause imaginary and complex integer datatypes and operations on values of an imaginary or complex integer datatype are specified.

A complex integer datatype is constructed from an integer datatype. There should be at least one imaginary integer datatype and at least one complex integer datatype for each provided integer datatype.

I is the set of non-special values, $I \subseteq \mathcal{Z}$, for an integer datatype conforming to part 1. Integer datatypes conforming to part 1 often do not contain any NaN or infinity values, even though they may do so. Therefore this clause has no specifications for such values as arguments or results.

$\mathrm{i}(I)$ (see Clause 4.1.6) is the set of non-special values in an imaginary integer datatype, constructed from the integer datatype corresponding to non-special value set I.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$\mathrm{c}(I)$ (see Clause 4.1.6) is the set of non-special values in a complex integer datatype, constructed from the integer datatype corresponding to the non-special value set I.

NOTE – The operations that return zero for certain cases, according to the specifications below, may in a subset of those cases return negative zero instead, if negative zero can be represented. Compare the specifications for corresponding complex floating point operations in Clause 5.2.

5.1.1 The complex integer *result* helper function

The $result_{\mathrm{c}(I)}$ helper function:

$$result_{\mathrm{c}(I)} : \mathcal{G} \rightarrow \mathrm{c}(I) \cup \{\textbf{overflow}\}$$

$$result_{\mathrm{c}(I)}(z) \quad = result_I(\mathfrak{Re}(z)) + \hat{\imath}\cdot\, result_I(\mathfrak{Im}(z))$$

NOTE – If one or both of the $result_I$ (defined in part 2) function applications on the right side returns **overflow**, then the $result_{\mathrm{c}(I)}$ application returns **overflow**. The continuation values used when **overflow** occurs are to be specified by the binding or implementation. The same holds also for other exceptional values and also for the specifications below that do not use $result_{\mathrm{c}(I)}$ but specify the result parts directly.

5.1.2 Imaginary and complex integer operations

5.1.2.1 Complex integer comparisons

$$eq_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$$

$$eq_{\mathrm{i}(I)}(\hat{\imath}\cdot\, y, \hat{\imath}\cdot\, w) \quad = eq_I(y, w)$$

$$eq_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$$

$$eq_{I,\mathrm{i}(I)}(x, \hat{\imath}\cdot\, w) \quad = eq_{\mathrm{c}(I)}(x + \hat{\imath}\cdot\, 0, 0 + \hat{\imath}\cdot\, w)$$

$$eq_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \textbf{Boolean}$$

$$eq_{\mathrm{i}(I),I}(\hat{\imath}\cdot\, y, z) \quad = eq_{\mathrm{c}(I)}(0 + \hat{\imath}\cdot\, y, z + \hat{\imath}\cdot\, 0)$$

$$eq_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \textbf{Boolean}$$

$$eq_{I,\mathrm{c}(I)}(x, z + \hat{\imath}\cdot\, w) \quad = eq_{\mathrm{c}(I)}(x + \hat{\imath}\cdot\, 0, z + \hat{\imath}\cdot\, w)$$

$$eq_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \textbf{Boolean}$$

$$eq_{\mathrm{c}(I),I}(x + \hat{\imath}\cdot\, y, z) \quad = eq_{\mathrm{c}(I)}(x + \hat{\imath}\cdot\, y, z + \hat{\imath}\cdot\, 0)$$

$$eq_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \textbf{Boolean}$$

$$eq_{\mathrm{i}(I),\mathrm{c}(I)}(\hat{\imath}\cdot\, y, z + \hat{\imath}\cdot\, w) \quad = eq_{\mathrm{c}(I)}(0 + \hat{\imath}\cdot\, y, z + \hat{\imath}\cdot\, w)$$

$$eq_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$eq_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = eq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, 0 \mathbf{+\hat{\imath}\cdot} w)$$

$$eq_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathbf{Boolean}$$

$$\begin{aligned} eq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) &= \mathbf{true} && \text{if } eq_I(x,z) = \mathbf{true} \text{ and } eq_I(y,w) = \mathbf{true} \\ &= \mathbf{false} && \text{if } eq_I(x,z) = \mathbf{false} \text{ and } eq_I(y,w) = \mathbf{true} \\ &= \mathbf{false} && \text{if } eq_I(x,z) = \mathbf{true} \text{ and } eq_I(y,w) = \mathbf{false} \\ &= \mathbf{false} && \text{if } eq_I(x,z) = \mathbf{false} \text{ and } eq_I(y,w) = \mathbf{false} \end{aligned}$$

$$neq_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = neq_I(y, w)$$

$$neq_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \mathbf{Boolean}$$

$$neq_{I,\mathrm{i}(I)}(x, \mathbf{\hat{\imath}\cdot} w) = neq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} 0, 0 \mathbf{+\hat{\imath}\cdot} w)$$

$$neq_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{i}(I),I}(\mathbf{\hat{\imath}\cdot} y, z) = neq_{\mathrm{c}(I)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} 0)$$

$$neq_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathbf{Boolean}$$

$$neq_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) = neq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)$$

$$neq_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) = neq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} 0)$$

$$neq_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = neq_{\mathrm{c}(I)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)$$

$$neq_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = neq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, 0 \mathbf{+\hat{\imath}\cdot} w)$$

$$neq_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathbf{Boolean}$$

$$\begin{aligned} neq_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) &= \mathbf{true} && \text{if } neq_I(x,z) = \mathbf{true} \text{ and } neq_I(y,w) = \mathbf{true} \\ &= \mathbf{true} && \text{if } neq_I(x,z) = \mathbf{false} \text{ and } neq_I(y,w) = \mathbf{true} \\ &= \mathbf{true} && \text{if } neq_I(x,z) = \mathbf{true} \text{ and } neq_I(y,w) = \mathbf{false} \\ &= \mathbf{false} && \text{if } neq_I(x,z) = \mathbf{false} \text{ and } neq_I(y,w) = \mathbf{false} \end{aligned}$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$lss_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$
$lss_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) \ = lss_I(y, w)$

$leq_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$
$leq_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) \ = leq_I(y, w)$

$gtr_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$
$gtr_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) \ = gtr_I(y, w)$

$geq_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$
$geq_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = geq_I(y, w)$

5.1.2.2 Multiplication by the imaginary unit

$itimes_{I\rightarrow\mathrm{i}(I)} : I \rightarrow \mathrm{i}(I)$
$itimes_{I\rightarrow\mathrm{i}(I)}(x) \ = \hat{\mathbf{i}}\cdot x$

$itimes_{\mathrm{i}(I)\rightarrow I} : \mathrm{i}(I) \rightarrow I \cup \{\textbf{overflow}\}$
$itimes_{\mathrm{i}(I)\rightarrow I}(\hat{\mathbf{i}}\cdot y)$
$= neg_I(y)$

$itimes_{\mathrm{c}(I)} : \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\textbf{overflow}\}$
$itimes_{\mathrm{c}(I)}(x + \hat{\mathbf{i}}\cdot y)$
$= neg_I(y) + \hat{\mathbf{i}}\cdot x$

5.1.2.3 The real and imaginary parts of a complex value

$re_I : I \rightarrow I$
$re_I(x) \quad = x \qquad \text{if } x \in I$

$re_{\mathrm{i}(I)} : \mathrm{i}(I) \rightarrow \{0\}$
$re_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y) \quad = 0 \qquad \text{if } y \in I$

$re_{\mathrm{c}(I)} : \mathrm{c}(I) \rightarrow I$
$re_{\mathrm{c}(I)}(x + \hat{\mathbf{i}}\cdot y) \quad = x \qquad \text{if } x \in I$

$im_I : I \rightarrow \{0\}$
$im_I(x) \quad = 0 \qquad \text{if } x \in I$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$im_{\mathrm{i}(I)} : \mathrm{i}(I) \rightarrow I$
$im_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot\, y) \quad = y \qquad\qquad$ if $y \in I$

$im_{\mathrm{c}(I)} : \mathrm{c}(I) \rightarrow I$
$im_{\mathrm{c}(I)}(x + \hat{\mathbf{i}}\cdot\, y) \quad = y \qquad\qquad$ if $y \in I$

5.1.2.4 Formation of a complex integer from two real valued integers

$plusitimes_I : I \times I \rightarrow \mathrm{c}(I)$
$plusitimes_I(x, z)$
$= x + \hat{\mathbf{i}}\cdot\, z$

5.1.2.5 Basic complex integer arithmetic

$neg_{\mathrm{i}(I)} : \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}\}$
$neg_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot\, y) \quad = \hat{\mathbf{i}}\cdot\, neg_I(y)$

$neg_{\mathrm{c}(I)} : \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$
$neg_{\mathrm{c}(I)}(x + \hat{\mathbf{i}}\cdot\, y) \quad = neg_I(x) + \hat{\mathbf{i}}\cdot\, neg_I(y)$

$conj_I : I \rightarrow I$
$conj_I(x) \quad = x$

$conj_{\mathrm{i}(I)} : \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}\}$
$conj_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot\, y) \quad = \hat{\mathbf{i}}\cdot\, neg_I(y)$

$conj_{\mathrm{c}(I)} : \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$
$conj_{\mathrm{c}(I)}(x + \hat{\mathbf{i}}\cdot\, y)$
$= x + \hat{\mathbf{i}}\cdot\, neg_I(y)$

$add_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}\}$
$add_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot\, y, \hat{\mathbf{i}}\cdot\, w) = \hat{\mathbf{i}}\cdot\, add_I(y, w)$

$add_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \mathrm{c}(I)$
$add_{I,\mathrm{i}(I)}(x, \hat{\mathbf{i}}\cdot\, w) \quad = x + \hat{\mathbf{i}}\cdot\, w$

$add_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \mathrm{c}(I)$
$add_{\mathrm{i}(I),I}(\hat{\mathbf{i}}\cdot\, y, z) \quad = z + \hat{\mathbf{i}}\cdot\, y$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$add_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$add_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) = add_I(x, z) \mathbf{+\hat{\imath}\cdot} w$$

$$add_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$add_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) = add_I(x, z) \mathbf{+\hat{\imath}\cdot} y$$

$$add_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$add_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = z \mathbf{+\hat{\imath}\cdot} add_I(y, w)$$

$$add_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$add_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} add_I(y, w)$$

$$add_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$add_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = add_I(x, z) \mathbf{+\hat{\imath}\cdot} add_I(y, w)$$

$$sub_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} sub_I(y, w)$$

$$sub_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{I,\mathrm{i}(I)}(x, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} neg_I(w)$$

$$sub_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{\mathrm{i}(I),I}(\mathbf{\hat{\imath}\cdot} y, z) = neg_I(z) \mathbf{+\hat{\imath}\cdot} y$$

$$sub_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) = sub_I(x, z) \mathbf{+\hat{\imath}\cdot} neg_I(w)$$

$$sub_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) = sub_I(x, z) \mathbf{+\hat{\imath}\cdot} y$$

$$sub_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = neg_I(z) \mathbf{+\hat{\imath}\cdot} sub_I(y, w)$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$sub_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} sub_I(y, w)$$

$$sub_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$sub_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = sub_I(x, z) \mathbf{+\hat{\imath}\cdot} sub_I(y, w)$$

$$mul_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow I \cup \{\mathbf{overflow}\}$$
$$mul_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = result_I(-(y \cdot w)) \qquad \text{if } y, w \in I$$

$$mul_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{I,\mathrm{i}(I)}(x, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} mul_I(x, w)$$

$$mul_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{\mathrm{i}(I),I}(\mathbf{\hat{\imath}\cdot} y, z) = \mathbf{\hat{\imath}\cdot} mul_I(y, z)$$

$$mul_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) = mul_I(x, z) \mathbf{+\hat{\imath}\cdot} mul_I(x, w)$$

$$mul_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) = mul_I(x, z) \mathbf{+\hat{\imath}\cdot} mul_I(y, z)$$

$$mul_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = result_I(-(y \cdot w)) \mathbf{+\hat{\imath}\cdot} mul_I(y, z) \qquad \text{if } y, z, w \in I$$

$$mul_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = result_I(-(y \cdot w)) \mathbf{+\hat{\imath}\cdot} mul_I(x, w) \qquad \text{if } x, y, w \in I$$

$$mul_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}\}$$
$$mul_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = result_{\mathrm{c}(I)}((x + (\tilde{\imath} \cdot y)) \cdot (z + (\tilde{\imath} \cdot w))) \qquad \text{if } x, y, z, w \in I$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

5.1.2.6 Absolute value and signum of integers and imaginary integers

NOTE – abs_I is specified in part 1.

$abs_{\mathrm{i}(I)} : \mathrm{i}(I) \rightarrow I \cup \{\textbf{overflow}\}$

$abs_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y) \quad = abs_I(y)$

$signum_I : I \rightarrow \{-1, 1\}$

$signum_I(x) \quad = 1 \quad$ if $(x \in I$ and $x \geqslant 0)$

$\qquad\qquad\quad = -1 \quad$ if $(x \in I$ and $x < 0)$

$signum_{\mathrm{i}(I)} : \mathrm{i}(I) \rightarrow \{\hat{\mathbf{i}}\cdot(-1), \hat{\mathbf{i}}\cdot 1\}$

$signum_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y) = \hat{\mathbf{i}}\cdot signum_I(y)$

5.1.2.7 Divisibility interrogation

$divides_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$

$divides_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w)$

$\qquad = divides_I(y, w)$

$divides_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$

$divides_{I,\mathrm{i}(I)}(x, \hat{\mathbf{i}}\cdot w)$

$\qquad = divides_I(x, w)$

$divides_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \textbf{Boolean}$

$divides_{\mathrm{i}(I),I}(\hat{\mathbf{i}}\cdot y, z)$

$\qquad = divides_I(y, z)$

$divides_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \textbf{Boolean}$

$divides_{I,\mathrm{c}(I)}(x, z \mathbf{+} \hat{\mathbf{i}}\cdot w)$

$\qquad = divides_{\mathrm{c}(I)}(x \mathbf{+} \hat{\mathbf{i}}\cdot 0, z \mathbf{+} \hat{\mathbf{i}}\cdot w)$

$divides_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \textbf{Boolean}$

$divides_{\mathrm{c}(I),I}(x \mathbf{+} \hat{\mathbf{i}}\cdot y, z)$

$\qquad = divides_{\mathrm{c}(I)}(x \mathbf{+} \hat{\mathbf{i}}\cdot y, z \mathbf{+} \hat{\mathbf{i}}\cdot 0)$

$divides_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \textbf{Boolean}$

$divides_{\mathrm{i}(I),\mathrm{c}(I)}(\hat{\mathbf{i}}\cdot y, z \mathbf{+} \hat{\mathbf{i}}\cdot w)$

$\qquad = divides_{\mathrm{c}(I)}(0 \mathbf{+} \hat{\mathbf{i}}\cdot y, z \mathbf{+} \hat{\mathbf{i}}\cdot w)$

$divides_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \textbf{Boolean}$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$divides_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = divides_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, 0 \mathbf{+\hat{\imath}\cdot} w)$$

$$divides_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathbf{Boolean}$$

$$divides_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) \begin{cases} = \mathbf{true} & \text{if } x, y, z, w \in I \text{ and } (x + (\tilde{\imath} \cdot y)) \mid (z + (\tilde{\imath} \cdot w)) \\ = \mathbf{false} & \text{if } x, y, z, w \in I \text{ and not } (x + (\tilde{\imath} \cdot y)) \mid (z + (\tilde{\imath} \cdot w)) \end{cases}$$

5.1.2.8 Integer division and remainder extended to imaginary and complex integers

For these operations, I shall be signed.

NOTE – Even though the integer division operations in principle could be included also for unsigned integer datatypes, that would result in operations that would overflow (mathematically have negative subresults) for so many argument values, that the inclusion of those operations would be pointless. The same goes for the remainder operations.

$$quot_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow I \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

$$quot_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = quot_I(y, w)$$

$$quot_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

$$quot_{I,\mathrm{i}(I)}(x, \mathbf{\hat{\imath}\cdot} w) \begin{cases} = \mathbf{\hat{\imath}\cdot}\, minint_I & \text{if } x = minint_I \text{ and } w = -1 \\ = \mathbf{\hat{\imath}\cdot}\, neg_I(group_I(x, w)) & \text{otherwise} \end{cases}$$

$$quot_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

$$quot_{\mathrm{i}(I),I}(\mathbf{\hat{\imath}\cdot} y, z) = \mathbf{\hat{\imath}\cdot}\, quot_I(y, z)$$

$$quot_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

$$quot_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) = quot_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)$$

$$quot_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

$$quot_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) = quot_I(x, z) \mathbf{+\hat{\imath}\cdot}\, quot_I(y, z)$$

$$quot_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

$$quot_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = quot_{\mathrm{c}(I)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)$$

$$quot_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

$$
\begin{aligned}
&quot_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = neg_I(y) \mathbf{+\hat{\imath}\cdot} minint_I \qquad \text{if } x = minint_I \text{ and } w = -1 \\
&\qquad = quot_I(y, w) \mathbf{+\hat{\imath}\cdot} neg_I(group_I(x, w)) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&quot_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&quot_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = result_{\mathrm{c}(I)}(\lfloor (x + (\tilde{\imath} \cdot y))/(z + (\tilde{\imath} \cdot w)) \rfloor) \\
&\qquad\qquad\qquad\qquad \text{if } x, y, z, w \in I \text{ and } z + (\tilde{\imath} \cdot w) \neq 0 \\
&\qquad = quot_I(x, 0) \mathbf{+\hat{\imath}\cdot} quot_I(y, 0) \\
&\qquad\qquad\qquad\qquad \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&mod_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \to \mathrm{i}(I) \cup \{\mathbf{invalid}\} \\
&mod_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = \mathbf{\hat{\imath}\cdot} mod_I(y, w)
\end{aligned}
$$

$$
\begin{aligned}
&mod_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \to I \cup \{\mathbf{invalid}\} \\
&mod_{I,\mathrm{i}(I)}(x, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = neg_I(pad_I(x, w))
\end{aligned}
$$

$$
\begin{aligned}
&mod_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \to \mathrm{i}(I) \cup \{\mathbf{invalid}\} \\
&mod_{\mathrm{i}(I),I}(\mathbf{\hat{\imath}\cdot} y, z) = \mathbf{\hat{\imath}\cdot} mod_I(y, z)
\end{aligned}
$$

$$
\begin{aligned}
&mod_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{invalid}\} \\
&mod_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = mod_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)
\end{aligned}
$$

$$
\begin{aligned}
&mod_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \to \mathrm{c}(I) \cup \{\mathbf{invalid}\} \\
&mod_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) \\
&\qquad = mod_I(x, z) \mathbf{+\hat{\imath}\cdot} mod_I(y, z)
\end{aligned}
$$

$$
\begin{aligned}
&mod_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{invalid}\} \\
&mod_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = mod_{\mathrm{c}(I)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)
\end{aligned}
$$

$$
\begin{aligned}
&mod_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \to \mathrm{c}(I) \cup \{\mathbf{invalid}\} \\
&mod_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = neg_I(pad_I(x, w)) \mathbf{+\hat{\imath}\cdot} mod_I(y, w)
\end{aligned}
$$

$$
mod_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{invalid}\}
$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$
\begin{aligned}
&mod_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = result_{\mathrm{c}(I)}((x + (\tilde{\imath} \cdot y)) - (\lfloor (x + (\tilde{\imath} \cdot y))/(z + (\tilde{\imath} \cdot w)) \rfloor \cdot (z + (\tilde{\imath} \cdot w)))) \\
&\qquad\qquad\qquad\qquad \text{if } x, y, z, w \in I \text{ and } z + (\tilde{\imath} \cdot w) \neq 0 \\
&\qquad = \mathbf{invalid}(\mathbf{qNaN} \mathbf{+\hat{\imath}\cdot} \mathbf{qNaN}) \\
&\qquad\qquad\qquad\qquad \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \to I \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = ratio_I(y, w)
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \to \mathrm{i}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{I,\mathrm{i}(I)}(x, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = \mathbf{\hat{\imath}\cdot}\, minint_I \qquad \text{if } x = minint_I \text{ and } w = -1 \\
&\qquad = \mathbf{\hat{\imath}\cdot}\, neg_I(ratio_I(x, w)) \qquad \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \to \mathrm{i}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{\mathrm{i}(I),I}(\mathbf{\hat{\imath}\cdot} y, z) = \mathbf{\hat{\imath}\cdot}\, ratio_I(y, z)
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = ratio_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) \\
&\qquad = ratio_I(x, z) \mathbf{+\hat{\imath}\cdot}\, ratio_I(y, z)
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = ratio_{\mathrm{c}(I)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) \\
&\qquad = neg_I(y) \mathbf{+\hat{\imath}\cdot}\, minint_I \qquad \text{if } x = minint_I \text{ and } w = -1 \\
&\qquad = ratio_I(y, w) \mathbf{+\hat{\imath}\cdot}\, neg_I(ratio_I(x, w)) \\
&\qquad\qquad\qquad\qquad \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&ratio_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&ratio_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) \\
&\qquad = result_{\mathrm{c}(I)}(\mathrm{round}((x + (\tilde{\imath} \cdot y))/(z + (\tilde{\imath} \cdot w)))) \\
&\qquad\qquad\qquad\qquad \text{if } x, y, z, w \in I \text{ and } z + (\tilde{\imath} \cdot w) \neq 0 \\
&\qquad = ratio_I(x, 0) \mathbf{+\hat{\imath}\cdot}\, ratio_I(y, 0) \\
&\qquad\qquad\qquad\qquad \text{otherwise}
\end{aligned}
$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$residue_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = \hat{\mathbf{i}}\cdot residue_I(y, w)$$

$$residue_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow I \cup \{\mathbf{invalid}\}$$
$$residue_{I,\mathrm{i}(I)}(x, \hat{\mathbf{i}}\cdot w) = residue_I(x, w)$$

$$residue_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \rightarrow \mathrm{i}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{\mathrm{i}(I),I}(\hat{\mathbf{i}}\cdot y, z) = \hat{\mathbf{i}}\cdot residue_I(y, z)$$

$$residue_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{I,\mathrm{c}(I)}(x, z \mathbf{+}\hat{\mathbf{i}}\cdot w) = residue_{\mathrm{c}(I)}(x \mathbf{+}\hat{\mathbf{i}}\cdot 0, z \mathbf{+}\hat{\mathbf{i}}\cdot w)$$

$$residue_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{\mathrm{c}(I),I}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, z) = residue_I(x, z) \mathbf{+}\hat{\mathbf{i}}\cdot residue_I(y, z)$$

$$residue_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{\mathrm{i}(I),\mathrm{c}(I)}(\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w) = residue_{\mathrm{c}(I)}(0 \mathbf{+}\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w)$$

$$residue_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = residue_I(x, w) \mathbf{+}\hat{\mathbf{i}}\cdot residue_I(y, w)$$

$$residue_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$residue_{\mathrm{c}(I)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w)$$
$$= result_{\mathrm{c}(I)}((x + (\tilde{\imath}\cdot y)) - (\mathrm{round}((x + (\tilde{\imath}\cdot y))/(z + (\tilde{\imath}\cdot w)))\cdot(z + (\tilde{\imath}\cdot w)))) \quad \text{if } x, y, z, w \in I \text{ and } z + (\tilde{\imath}\cdot w) \neq 0$$
$$= \mathbf{invalid}(\mathbf{qNaN} \mathbf{+}\hat{\mathbf{i}}\cdot \mathbf{qNaN}) \quad \text{otherwise}$$

$$group_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow I \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$
$$group_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = group_I(y, w)$$

$$group_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \rightarrow \mathrm{i}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$
\begin{aligned}
&group_{I,\mathrm{i}(I)}(x, \hat{\mathbf{i}}\cdot w) \\
&\qquad = \hat{\mathbf{i}}\cdot minint_I && \text{if } x = minint_I \text{ and } w = -1 \\
&\qquad = \hat{\mathbf{i}}\cdot neg_I(quot_I(x, w)) && \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&group_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \to \mathrm{i}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&group_{\mathrm{i}(I),I}(\hat{\mathbf{i}}\cdot y, z) \\
&\qquad = \hat{\mathbf{i}}\cdot group_I(y, z)
\end{aligned}
$$

$$
\begin{aligned}
&group_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&group_{I,\mathrm{c}(I)}(x, z \mathbf{+}\hat{\mathbf{i}}\cdot w) \\
&\qquad = group_{\mathrm{c}(I)}(x \mathbf{+}\hat{\mathbf{i}}\cdot 0, z \mathbf{+}\hat{\mathbf{i}}\cdot w)
\end{aligned}
$$

$$
\begin{aligned}
&group_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&group_{\mathrm{c}(I),I}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, z) \\
&\qquad = group_I(x, z) \mathbf{+}\hat{\mathbf{i}}\cdot group_I(y, z)
\end{aligned}
$$

$$
\begin{aligned}
&group_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&group_{\mathrm{i}(I),\mathrm{c}(I)}(\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w) \\
&\qquad = group_{\mathrm{c}(I)}(0 \mathbf{+}\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w)
\end{aligned}
$$

$$
\begin{aligned}
&group_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&group_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) \\
&\qquad = neg_I(y) \mathbf{+}\hat{\mathbf{i}}\cdot minint_I && \text{if } x = minint_I \text{ and } w = -1 \\
&\qquad = group_I(y, w) \mathbf{+}\hat{\mathbf{i}}\cdot neg_I(quot_I(x, w)) \\
&\qquad\qquad && \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&group_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \to \mathrm{c}(I) \cup \{\mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\} \\
&group_{\mathrm{c}(I)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w) \\
&\qquad = result_{\mathrm{c}(I)}(\lceil (x + (\tilde{\imath}\cdot y))/(z + (\tilde{\imath}\cdot w))\rceil) \\
&\qquad\qquad && \text{if } x, y, z, w \in I \text{ and } z + (\tilde{\imath}\cdot w) \neq 0 \\
&\qquad = group_I(x, 0) \mathbf{+}\hat{\mathbf{i}}\cdot group_I(y, 0) \\
&\qquad\qquad && \text{otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&pad_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \to \mathrm{i}(I) \cup \{\mathbf{invalid}\} \\
&pad_{\mathrm{i}(I)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = \hat{\mathbf{i}}\cdot pad_I(y, w)
\end{aligned}
$$

$$
\begin{aligned}
&pad_{I,\mathrm{i}(I)} : I \times \mathrm{i}(I) \to I \cup \{\mathbf{invalid}\} \\
&pad_{I,\mathrm{i}(I)}(x, \hat{\mathbf{i}}\cdot w) = neg_I(mod_I(x, w))
\end{aligned}
$$

$$
\begin{aligned}
&pad_{\mathrm{i}(I),I} : \mathrm{i}(I) \times I \to \mathrm{i}(I) \cup \{\mathbf{invalid}\} \\
&pad_{\mathrm{i}(I),I}(\hat{\mathbf{i}}\cdot y, z) = \hat{\mathbf{i}}\cdot pad_I(y, z)
\end{aligned}
$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$pad_{I,\mathrm{c}(I)} : I \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$pad_{I,\mathrm{c}(I)}(x, z \mathbf{+\hat{\imath}\cdot} w) = pad_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)$$

$$pad_{\mathrm{c}(I),I} : \mathrm{c}(I) \times I \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$pad_{\mathrm{c}(I),I}(x \mathbf{+\hat{\imath}\cdot} y, z) = pad_I(x, z) \mathbf{+\hat{\imath}\cdot} pad_I(y, z)$$

$$pad_{\mathrm{i}(I),\mathrm{c}(I)} : \mathrm{i}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$pad_{\mathrm{i}(I),\mathrm{c}(I)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = pad_{\mathrm{c}(I)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)$$

$$pad_{\mathrm{c}(I),\mathrm{i}(I)} : \mathrm{c}(I) \times \mathrm{i}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$pad_{\mathrm{c}(I),\mathrm{i}(I)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = neg_I(mod_I(x, w)) \mathbf{+\hat{\imath}\cdot} pad_I(y, w)$$

$$pad_{\mathrm{c}(I)} : \mathrm{c}(I) \times \mathrm{c}(I) \rightarrow \mathrm{c}(I) \cup \{\mathbf{invalid}\}$$
$$pad_{\mathrm{c}(I)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)$$
$$= result_{\mathrm{c}(I)}((\lceil (x + (\tilde{\imath} \cdot y))/(z + (\tilde{\imath} \cdot w)) \rceil \cdot (z + (\tilde{\imath} \cdot w))) - (x + (\tilde{\imath} \cdot y))) \quad \text{if } x, y, z, w \in I \text{ and } z + (\tilde{\imath} \cdot w) \neq 0$$
$$= \mathbf{invalid}(\mathbf{qNaN} \mathbf{+\hat{\imath}\cdot} \mathbf{qNaN}) \quad \text{otherwise}$$

5.1.2.9 Maximum and minimum

$$max_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \mathrm{i}(I)$$
$$max_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} max_I(y, w)$$

$$min_{\mathrm{i}(I)} : \mathrm{i}(I) \times \mathrm{i}(I) \rightarrow \mathrm{i}(I)$$
$$min_{\mathrm{i}(I)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} min_I(y, w)$$

$$max_seq_{\mathrm{i}(I)} : [\mathrm{i}(I)] \rightarrow \mathrm{i}(I) \cup \{\mathbf{infinitary}\}$$
$$max_seq_{\mathrm{i}(I)}([\mathbf{\hat{\imath}\cdot} y_1, ..., \mathbf{\hat{\imath}\cdot} y_n]) = \mathbf{\hat{\imath}\cdot} max_seq_I([y_1, ..., y_n])$$

$$min_seq_{\mathrm{i}(I)} : [\mathrm{i}(I)] \rightarrow \mathrm{i}(I) \cup \{\mathbf{infinitary}\}$$
$$min_seq_{\mathrm{i}(I)}([\mathbf{\hat{\imath}\cdot} y_1, ..., \mathbf{\hat{\imath}\cdot} y_n]) = \mathbf{\hat{\imath}\cdot} min_seq_I([y_1, ..., y_n])$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

5.2 Imaginary and complex floating point datatypes and operations

Clause 5.2 of part 1 and Clause 5.2 of part 2 of ISO/IEC 10967 specify floating point datatypes and a number of operations on values of a floating point datatype. In this clause imaginary and complex floating point datatypes and operations on values of an imaginary or complex floating point datatype are specified.

NOTE – Further operations on values of an imaginary or complex floating point datatype, for elementary complex floating point numerical functions, are specified in Clause 5.3.

An imaginary or complex floating point datatype is constructed from a floating point datatype. There should be at least one imaginary floating point datatype and at least one complex floating point datatype for each provided floating point datatype.

F is the non-special value set, $F \subset \mathcal{R}$, for a floating point datatype conforming to part 1 of ISO/IEC 10967. Floating point datatypes conforming to part 1 often do contain $\mathbf{-0}$, infinity, and NaN values. Therefore, in this clause there are specifications for such values as arguments.

$\mathrm{i}(F)$ (see Clause 4.1.6) is the set of non-special values in an imaginary floating point datatype, constructed from the floating point datatype corresponding to the non-special value set F.

$\mathrm{c}(F)$ (see Clause 4.1.6) is the set of non-special values in a complex floating point datatype, constructed from the floating point datatype corresponding to the non-special value set F.

5.2.1 Maximum error requirements

Some of the operations are exact, such as taking the imaginary part of a complex value. Some operations are approximate, with maximum error requirements implied by their exact relationships with operations defined in part 1, part 2, or this part of this International Standard. Some other approximate operations have new error parameters associated with them as detailed in the specifications.

The approximation helper functions for the individual operations in these subclauses have maximum error parameters that describe the maximum *relative* error, in ulps, of the helper function composed with $nearest_F$, for non-subnormal results. The maximum error parameters also describe the maximum *absolute* error, in ulps, for $-fminN_F$, $fminN_F$, subnormal results and underflow continuation values if $denorm_F = \mathbf{true}$. All maximum error parameters shall have a value that is $\geqslant 0.5$. For the maximum value of the maximum error parameters, see the specification of each of the maximum error parameters. See also Annex A, on partial conformity. The relevant maximum error parameters shall be made available to programs.

When the maximum error for an approximation helper function $h_{\mathrm{c}(F)}$, approximating f, is $max_error_op_{\mathrm{c}(F)}$, then for all arguments $z, ... \in \mathcal{C}_F \times ...$ the following equations shall hold:

$$|\mathfrak{Re}(f(z,...)) - nearest_F(\mathfrak{Re}(h_{\mathrm{c}(F)}(z,...)))| \leqslant max_error_op_{\mathrm{c}(F)} \cdot r_F^{e_F(\mathfrak{Re}(f(z,...)))-p_F}$$
$$|\mathfrak{Im}(f(z,...)) - nearest_F(\mathfrak{Im}(h_{\mathrm{c}(F)}(z,...)))| \leqslant max_error_op_{\mathrm{c}(F)} \cdot r_F^{e_F(\mathfrak{Im}(f(z,...)))-p_F}$$

Some operations have a **Boolean** parameter $box_error_mode_op_{\mathrm{c}(F)}$. If such a parameter is present for an operation and that parameter has the value **true**, then the sign requirements need not be fulfilled (see below) and the maximum error requirements are modified to the following:

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

$$|\mathfrak{Re}(f(z,...)) - nearest_F(\mathfrak{Re}(h_{c(F)}(z,...)))| \leqslant max_error_op_{c(F)} \cdot r_F^{e_F(|f(z,...)|)-p_F}$$
$$|\mathfrak{Im}(f(z,...)) - nearest_F(\mathfrak{Im}(h_{c(F)}(z,...)))| \leqslant max_error_op_{c(F)} \cdot r_F^{e_F(|f(z,...)|)-p_F}$$

The default value for a $box_error_mode_op_{c(F)}$ parameter should be **false**.

NOTES

1 Partially conforming implementations may have greater values for maximum error parameters than stipulated below. See Annex A.

2 Multiplication and division of complex values have the box error mode parameter. See 5.2.6.

3 The relative error requirement results in a 'rectangular' error bound, which can only span over a zero (in either dimension) when the returned result is very close to 0 (but see 'Sign requirements' below).

4 The box error requirement results in a 'square-formed' error bound, which for cancellation cases can span over zero in either or both dimensions. Relative error requirements in each axis can be fulfilled also for multiplication and division, but that is often considered too inefficient, and an implementation that may suffer from cancellation is often used instead.

5.2.2 Sign requirements

The following sign requirements shall hold:

a) The real or imaginary part of the result of an approximation helper function shall be zero exactly at the points where the real or imaginary part (respectively) of the approximated mathematical function is exactly zero.

b) At a point where an approximation helper function result part is not zero, the result part shall have the same sign as the corresponding result part of the approximated mathematical function at that point.

However, the following exceptions are made:

a) For the trigonometric helper functions, these zero and sign requirements are imposed only for arguments, $x + (\tilde{\imath} \cdot y)$, such that $|x| \leqslant big_angle_r_F$ (see Clause 5.3.2; $big_angle_r_F$ is specified in part 2).

b) If there is a $box_error_mode_op_{c(F)}$ parameter for an operation and that parameter has the value **true**, then the sign requirements are not imposed for that operation.

NOTES

1 For the operations, the continuation value after an **underflow** may be zero (including negative zero) as given by $result_F^*$ (see part 2), even though the approximation helper function is not zero at that point. Such zero results are required to be accompanied by an **underflow** notification. When appropriate, zero may also be returned for IEC 60559 infinities arguments. See the individual specifications.

2 Multiplication and division of complex values have modified sign requirements. See above and 5.2.6.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

5.2.3 Monotonicity requirements

For this document, each approximation helper function shall be a monotonic approximation, for each real and imaginary argument part individually, and the real and imaginary result parts individually, to the mathematical function it is approximating.

For the trigonometric approximation helper functions, the monotonic approximation requirement is imposed only for arguments, $x + (\tilde{\imath} \cdot y)$, such that $|x| \leqslant big_angle_r_F$ (see Clause 5.3.2). Similarly, for the exponentiation approximation helper functions, the monotonic approximation requirement is imposed only for arguments, $x + (\tilde{\imath} \cdot y)$, such that $|y| \leqslant big_angle_r_F$ (see Clause 5.3.1).

NOTES

1 As in part 2, the monotonicity requirement applies to each real dimension individually. For the complex operations, it thus applies to each real and imaginary part of the argument(s) individually, and individually to the real and imaginary parts of the result.

2 As in part 2, the monotonicity requirement applies individually to each monotonic interval of the approximated mathematical function.

3 There is an exact *relationship* between the trigonometric and hyperbolic *operations* specified in this document, and thus there are no complex hyperbolic approximation helper functions used in this document.

5.2.4 The complex floating point *result* helper functions

$$result^*_{\mathrm{c}(F)} : \mathcal{C} \times (\mathcal{R} \to F^*) \to \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$

$$\begin{aligned} &result^*_{\mathrm{c}(F)}(z, rnd) \\ &\qquad = result^*_F(\mathfrak{Re}(z), rnd) \mathbf{+\hat{\imath}\cdot} result^*_F(\mathfrak{Im}(z), rnd) \end{aligned}$$

NOTES

1 **overflow** and **underflow** can both occur for a single application of a complex operation. Likewise, e.g., one part may overflow, while the other produced a value in F. This is not exposed accurately by the mathematical framework used in this document. However, handling that accurately, would add complexity to the framework with little gain.

2 $result^*_F$ is defined in part 2.

The $no_result_{\mathrm{c}(F)}$, $no_result_{\mathrm{i}(F)}$, $no_result_{F \to \mathrm{c}(F)}$, $no_result_{\mathrm{i}(F) \to \mathrm{c}(F)}$, and $no_result2_{\mathrm{c}(F)}$ helper functions are defined as follows:

$$no_result_{\mathrm{c}(F)} : \mathrm{c}(F) \to \{\mathbf{invalid}\}$$

$$\begin{aligned} &no_result_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y) \\ &\qquad = \mathbf{invalid}(\mathbf{qNaN} \mathbf{+\hat{\imath}\cdot} \mathbf{qNaN}) && \text{if } x, y \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\ &\qquad = \mathbf{qNaN} \mathbf{+\hat{\imath}\cdot} \mathbf{qNaN} && \text{if at least one of } x \text{ and } y \text{ is a quiet NaN and neither is a signalling NaN} \\ &\qquad = \mathbf{invalid}(\mathbf{qNaN} \mathbf{+\hat{\imath}\cdot} \mathbf{qNaN}) && \text{if } x \text{ is a signalling NaN or } y \text{ is a signalling NaN} \end{aligned}$$

$$no_result_{\mathrm{i}(F)} : \mathrm{i}(F) \to \{\mathbf{invalid}\}$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$
\begin{aligned}
&no_result_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot y) \\
&\qquad = \mathbf{invalid}(\hat{\mathbf{i}}\cdot \mathbf{qNaN}) && \text{if } y \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\
&\qquad = \hat{\mathbf{i}}\cdot \mathbf{qNaN} && \text{if } y \text{ is a quiet NaN} \\
&\qquad = \mathbf{invalid}(\hat{\mathbf{i}}\cdot \mathbf{qNaN}) && \text{if } y \text{ is a signalling NaN}
\end{aligned}
$$

$$
\begin{aligned}
&no_result_{F\to\mathrm{c}(F)} : F \to \{\mathbf{invalid}\} \\
&no_result_{F\to\mathrm{c}(F)}(x) \\
&\qquad = no_result_{\mathrm{c}(F)}(x + \hat{\mathbf{i}}\cdot im_F(x))
\end{aligned}
$$

$$
\begin{aligned}
&no_result_{\mathrm{i}(F)\to\mathrm{c}(F)} : \mathrm{i}(F) \to \{\mathbf{invalid}\} \\
&no_result_{\mathrm{i}(F)\to\mathrm{c}(F)}(\hat{\mathbf{i}}\cdot y) \\
&\qquad = no_result_{\mathrm{c}(F)}(re_{\mathrm{i}(F)}(y) + \hat{\mathbf{i}}\cdot y)
\end{aligned}
$$

$$
\begin{aligned}
&no_result2_{\mathrm{c}(F)} : \mathrm{c}(F) \times \mathrm{c}(F) \to \{\mathbf{invalid}\} \\
&no_result2_{\mathrm{c}(F)}(x + \hat{\mathbf{i}}\cdot y, z + \hat{\mathbf{i}}\cdot w) \\
&\qquad = \mathbf{invalid}(\mathbf{qNaN} + \hat{\mathbf{i}}\cdot \mathbf{qNaN}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{if } x, y, z, w \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\
&\qquad = \mathbf{qNaN} + \hat{\mathbf{i}}\cdot \mathbf{qNaN} \qquad \text{if at least one of } x,\ y,\ z, \text{ and } w \text{ is a quiet NaN} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{and neither is a signalling NaN} \\
&\qquad = \mathbf{invalid}(\mathbf{qNaN} + \hat{\mathbf{i}}\cdot \mathbf{qNaN}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{if at least one of } x,\ y,\ z, \text{ and } w \text{ is a signalling NaN}
\end{aligned}
$$

These helper functions are used to specify both NaN argument handling and to handle non-NaN-argument cases where $\mathbf{invalid}(\mathbf{qNaN} + \hat{\mathbf{i}}\cdot \mathbf{qNaN})$ or $\mathbf{invalid}(\hat{\mathbf{i}}\cdot \mathbf{qNaN})$ is the appropriate result.

NOTES

3 The handling of other special values, if available, is left unspecified by this document. Other special values may be exact representations for π or e. At a lower level, such special values may be represented by IEC 60559 signalling NaNs, but which aren't regarded as NaNs at the LIA level.

4 $re_{\mathrm{i}(F)}$ and im_F are defined below.

5.2.5 Basic arithmetic for complex floating point

5.2.5.1 Complex floating point comparisons

$$
\begin{aligned}
&eq_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \to \mathbf{Boolean} \\
&eq_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) \quad = eq_F(y, w)
\end{aligned}
$$

$$
\begin{aligned}
&eq_{F,\mathrm{i}(F)} : F \times \mathrm{i}(F) \to \mathbf{Boolean} \\
&eq_{F,\mathrm{i}(F)}(x, \hat{\mathbf{i}}\cdot w) \quad = eq_{\mathrm{c}(F)}(x + \hat{\mathbf{i}}\cdot 0, 0 + \hat{\mathbf{i}}\cdot w)
\end{aligned}
$$

$$
\begin{aligned}
&eq_{\mathrm{i}(F),F} : \mathrm{i}(F) \times F \to \mathbf{Boolean} \\
&eq_{\mathrm{i}(F),F}(\hat{\mathbf{i}}\cdot y, z) \quad = eq_{\mathrm{c}(F)}(0 + \hat{\mathbf{i}}\cdot y, z + \hat{\mathbf{i}}\cdot 0)
\end{aligned}
$$

$$
eq_{F,\mathrm{c}(F)} : F \times \mathrm{c}(F) \to \mathbf{Boolean}
$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$eq_{F,\mathrm{c}(F)}(x, z \mathbf{+\hat{\imath}\cdot} w) = eq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)$$

$$eq_{\mathrm{c}(F),F} : \mathrm{c}(F) \times F \rightarrow \mathbf{Boolean}$$

$$eq_{\mathrm{c}(F),F}(x \mathbf{+\hat{\imath}\cdot} y, z) = eq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} 0)$$

$$eq_{\mathrm{i}(F),\mathrm{c}(F)} : \mathrm{i}(F) \times \mathrm{c}(F) \rightarrow \mathbf{Boolean}$$

$$eq_{\mathrm{i}(F),\mathrm{c}(F)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = eq_{\mathrm{c}(F)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w)$$

$$eq_{\mathrm{c}(F),\mathrm{i}(F)} : \mathrm{c}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$eq_{\mathrm{c}(F),\mathrm{i}(F)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = eq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y, 0 \mathbf{+\hat{\imath}\cdot} w)$$

$$eq_{\mathrm{c}(F)} : \mathrm{c}(F) \times \mathrm{c}(F) \rightarrow \mathbf{Boolean}$$

$$\begin{aligned} eq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) &= \mathbf{true} && \text{if } eq_F(x,z) = \mathbf{true} \text{ and } eq_F(y,w) = \mathbf{true} \\ &= \mathbf{false} && \text{if } eq_F(x,z) = \mathbf{false} \text{ and } eq_F(y,w) = \mathbf{true} \\ &= \mathbf{false} && \text{if } eq_F(x,z) = \mathbf{true} \text{ and } eq_F(y,w) = \mathbf{false} \\ &= \mathbf{false} && \text{if } eq_F(x,z) = \mathbf{false} \text{ and } eq_F(y,w) = \mathbf{false} \\ &= \mathbf{invalid}(\mathbf{false}) && \text{otherwise} \end{aligned}$$

$$neq_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = neq_F(y, w)$$

$$neq_{F,\mathrm{i}(F)} : F \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$neq_{F,\mathrm{i}(F)}(x, \mathbf{\hat{\imath}\cdot} w) = neq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} 0, 0 \mathbf{+\hat{\imath}\cdot} w)$$

$$neq_{\mathrm{i}(F),F} : \mathrm{i}(F) \times F \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{i}(F),F}(\mathbf{\hat{\imath}\cdot} y, z) = neq_{\mathrm{c}(F)}(0 \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} 0)$$

$$neq_{F,\mathrm{c}(F)} : F \times \mathrm{c}(F) \rightarrow \mathbf{Boolean}$$

$$neq_{F,\mathrm{c}(F)}(x, z \mathbf{+\hat{\imath}\cdot} w) = neq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} 0, z \mathbf{+\hat{\imath}\cdot} w)$$

$$neq_{\mathrm{c}(F),F} : \mathrm{c}(F) \times F \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{c}(F),F}(x \mathbf{+\hat{\imath}\cdot} y, z) = neq_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} 0)$$

$$neq_{\mathrm{i}(F),\mathrm{c}(F)} : \mathrm{i}(F) \times \mathrm{c}(F) \rightarrow \mathbf{Boolean}$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$neq_{\mathrm{i}(F),\mathrm{c}(F)}(\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w) = neq_{\mathrm{c}(F)}(0 \mathbf{+}\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w)$$

$$neq_{\mathrm{c}(F),\mathrm{i}(F)} : \mathrm{c}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$neq_{\mathrm{c}(F),\mathrm{i}(F)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = neq_{\mathrm{c}(F)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, 0 \mathbf{+}\hat{\mathbf{i}}\cdot w)$$

$$neq_{\mathrm{c}(F)} : \mathrm{c}(F) \times \mathrm{c}(F) \rightarrow \mathbf{Boolean}$$

$$\begin{aligned} neq_{\mathrm{c}(F)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y, z \mathbf{+}\hat{\mathbf{i}}\cdot w) & \\ &= \mathbf{true} && \text{if } neq_F(x,z) = \mathbf{true} \text{ and } neq_F(y,w) = \mathbf{true} \\ &= \mathbf{true} && \text{if } neq_F(x,z) = \mathbf{false} \text{ and } neq_F(y,w) = \mathbf{true} \\ &= \mathbf{true} && \text{if } neq_F(x,z) = \mathbf{true} \text{ and } neq_F(y,w) = \mathbf{false} \\ &= \mathbf{false} && \text{if } neq_F(x,z) = \mathbf{false} \text{ and } neq_F(y,w) = \mathbf{false} \\ &= \mathbf{invalid}(\mathbf{true}) && \text{otherwise} \end{aligned}$$

$$lss_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$lss_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = lss_F(y, w)$$

$$leq_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$leq_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = leq_F(y, w)$$

$$gtr_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$gtr_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = gtr_F(y, w)$$

$$geq_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathbf{Boolean}$$

$$geq_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot y, \hat{\mathbf{i}}\cdot w) = geq_F(y, w)$$

5.2.5.2 Multiplication by the imaginary unit

$$itimes_{F\rightarrow\mathrm{i}(F)} : F \rightarrow \mathrm{i}(F)$$

$$itimes_{F\rightarrow\mathrm{i}(F)}(x) = \hat{\mathbf{i}}\cdot x$$

$$itimes_{\mathrm{i}(F)\rightarrow F} : \mathrm{i}(F) \rightarrow F \cup \{\mathbf{-0}\}$$

$$itimes_{\mathrm{i}(F)\rightarrow F}(\hat{\mathbf{i}}\cdot y) = neg_F(y)$$

$$itimes_{\mathrm{c}(F)} : \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\})$$

$$itimes_{\mathrm{c}(F)}(x \mathbf{+}\hat{\mathbf{i}}\cdot y) = neg_F(y) \mathbf{+}\hat{\mathbf{i}}\cdot x$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

5.2.5.3 The real and imaginary parts of a complex value

$re_F : F \to F$

$$
\begin{aligned}
re_F(x) &= x && \text{if } x \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\
&= no_result_F(x) && \text{otherwise}
\end{aligned}
$$

$re_{\mathrm{i}(F)} : \mathrm{i}(F) \to \{-\mathbf{0}, 0\}$

$$
\begin{aligned}
re_{\mathrm{i}(F)}(\hat{\imath}\cdot y) &= 0 && \text{if } (y \in F \text{ and } y \geqslant 0) \text{ or } y = +\infty \\
&= -\mathbf{0} && \text{if } (y \in F \text{ and } y < 0) \text{ or } y \in \{-\infty, -\mathbf{0}\} \\
&= no_result_F(y) && \text{otherwise}
\end{aligned}
$$

$re_{\mathrm{c}(F)} : \mathrm{c}(F) \to F$

$$
\begin{aligned}
re_{\mathrm{c}(F)}(x + \hat{\imath}\cdot y) &= x && \text{if } x \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\
&= no_result_F(x) && \text{otherwise}
\end{aligned}
$$

$im_F : F \to \{-\mathbf{0}, 0\}$

$$
\begin{aligned}
im_F(x) &= -\mathbf{0} && \text{if } (x \in F \text{ and } x \geqslant 0) \text{ or } x = +\infty \\
&= 0 && \text{if } (x \in F \text{ and } x < 0) \text{ or } x \in \{-\infty, -\mathbf{0}\} \\
&= no_result_F(x) && \text{otherwise}
\end{aligned}
$$

$im_{\mathrm{i}(F)} : \mathrm{i}(F) \to F$

$$
\begin{aligned}
im_{\mathrm{i}(F)}(\hat{\imath}\cdot y) &= y && \text{if } y \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\
&= no_result_F(y) && \text{otherwise}
\end{aligned}
$$

$im_{\mathrm{c}(F)} : \mathrm{c}(F) \to F$

$$
\begin{aligned}
im_{\mathrm{c}(F)}(x + \hat{\imath}\cdot y) &= y && \text{if } y \in F \cup \{-\infty, -\mathbf{0}, +\infty\} \\
&= no_result_F(y) && \text{otherwise}
\end{aligned}
$$

5.2.5.4 Formation of a complex floating point from two floating point values

$plusitimes_F : F \times F \to \mathrm{c}(F)$

$$
\begin{aligned}
plusitimes_F(x, z) \\
&= x + \hat{\imath}\cdot z
\end{aligned}
$$

5.2.5.5 Fundamental complex floating point arithmetic

NOTE 1 – The addition and subtraction operations never underflow for an IEC 60559 implementation.

$neg_{\mathrm{i}(F)} : \mathrm{i}(F) \to \mathrm{i}(F \cup \{-\mathbf{0}\})$

$$
neg_{\mathrm{i}(F)}(\hat{\imath}\cdot y) = \hat{\imath}\cdot neg_F(y)
$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$neg_{\mathrm{c}(F)} : \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\})$$
$$neg_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y) = neg_F(x) \mathbf{+\hat{\imath}\cdot} neg_F(y)$$

$$conj_F : F \rightarrow F$$
$$conj_F(x) = x$$

$$conj_{\mathrm{i}(F)} : \mathrm{i}(F) \rightarrow \mathrm{i}(F \cup \{\mathbf{-0})\}$$
$$conj_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y) = \mathbf{\hat{\imath}\cdot} neg_F(y)$$

$$conj_{\mathrm{c}(F)} : \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\})$$
$$conj_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y) = x \mathbf{+\hat{\imath}\cdot} neg_F(y)$$

$$add_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathrm{i}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$add_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} add_F(y, w)$$

$$add_{F,\mathrm{i}(F)} : F \times \mathrm{i}(F) \rightarrow \mathrm{c}(F)$$
$$add_{F,\mathrm{i}(F)}(x, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} w$$

$$add_{\mathrm{i}(F),F} : \mathrm{i}(F) \times F \rightarrow \mathrm{c}(F)$$
$$add_{\mathrm{i}(F),F}(\mathbf{\hat{\imath}\cdot} y, z) = z \mathbf{+\hat{\imath}\cdot} y$$

$$add_{F,\mathrm{c}(F)} : F \times \mathrm{c}(F) \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$add_{F,\mathrm{c}(F)}(x, z \mathbf{+\hat{\imath}\cdot} w) = add_F(x, z) \mathbf{+\hat{\imath}\cdot} w$$

$$add_{\mathrm{c}(F),F} : \mathrm{c}(F) \times F \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$add_{\mathrm{c}(F),F}(x \mathbf{+\hat{\imath}\cdot} y, z) = add_F(x, z) \mathbf{+\hat{\imath}\cdot} y$$

$$add_{\mathrm{i}(F),\mathrm{c}(F)} : \mathrm{i}(F) \times \mathrm{c}(F) \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$add_{\mathrm{i}(F),\mathrm{c}(F)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = z \mathbf{+\hat{\imath}\cdot} add_F(y, w)$$

$$add_{\mathrm{c}(F),\mathrm{i}(F)} : \mathrm{c}(F) \times \mathrm{i}(F) \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$add_{\mathrm{c}(F),\mathrm{i}(F)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} add_F(y, w)$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$$add_{\mathrm{c}(F)} : \mathrm{c}(F) \times \mathrm{c}(F) \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$add_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = add_F(x, z) \mathbf{+\hat{\imath}\cdot} add_F(y, w)$$

$$sub_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow \mathrm{i}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$sub_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} sub_F(y, w)$$

$$sub_{F,\mathrm{i}(F)} : F \times \mathrm{i}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\})$$
$$sub_{F,\mathrm{i}(F)}(x, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} neg_F(w)$$

$$sub_{\mathrm{i}(F),F} : \mathrm{i}(F) \times F \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\})$$
$$sub_{\mathrm{i}(F),F}(\mathbf{\hat{\imath}\cdot} y, z) = neg_F(z) \mathbf{+\hat{\imath}\cdot} y$$

$$sub_{F,\mathrm{c}(F)} : F \times \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$sub_{F,\mathrm{c}(F)}(x, z \mathbf{+\hat{\imath}\cdot} w) = sub_F(x, z) \mathbf{+\hat{\imath}\cdot} neg_F(w)$$

$$sub_{\mathrm{c}(F),F} : \mathrm{c}(F) \times F \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$sub_{\mathrm{c}(F),F}(x \mathbf{+\hat{\imath}\cdot} y, z) = sub_F(x, z) \mathbf{+\hat{\imath}\cdot} y$$

$$sub_{\mathrm{i}(F),\mathrm{c}(F)} : \mathrm{i}(F) \times \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$sub_{\mathrm{i}(F),\mathrm{c}(F)}(\mathbf{\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = neg_F(z) \mathbf{+\hat{\imath}\cdot} sub_F(y, w)$$

$$sub_{\mathrm{c}(F),\mathrm{i}(F)} : \mathrm{c}(F) \times \mathrm{i}(F) \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$sub_{\mathrm{c}(F),\mathrm{i}(F)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = x \mathbf{+\hat{\imath}\cdot} sub_F(y, w)$$

$$sub_{\mathrm{c}(F)} : \mathrm{c}(F) \times \mathrm{c}(F) \rightarrow \mathrm{c}(F) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$sub_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y, z \mathbf{+\hat{\imath}\cdot} w) = sub_F(x, z) \mathbf{+\hat{\imath}\cdot} sub_F(y, w)$$

$$mul_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow F \cup \{\mathbf{-0}, \mathbf{underflow}, \mathbf{overflow}\}$$
$$mul_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w) = neg_F(mul_F(y, w))$$

$$mul_{F,\mathrm{i}(F)} : F \times \mathrm{i}(F) \rightarrow \mathrm{i}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$$
$$mul_{F,\mathrm{i}(F)}(x, \mathbf{\hat{\imath}\cdot} w) = \mathbf{\hat{\imath}\cdot} mul_F(x, w)$$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006

$mul_{\mathrm{i}(F),F} : \mathrm{i}(F) \times F \rightarrow \mathrm{i}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$
$mul_{\mathrm{i}(F),F}(\hat{\mathbf{i}}\cdot\, y, z) = \hat{\mathbf{i}}\cdot\, mul_F(y, z)$

$mul_{F,\mathrm{c}(F)} : F \times \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$
$mul_{F,\mathrm{c}(F)}(x, z \mathbf{+}\hat{\mathbf{i}}\cdot\, w)$
$\qquad = mul_F(x, z) \mathbf{+}\hat{\mathbf{i}}\cdot\, mul_F(x, w)$

$mul_{\mathrm{c}(F),F} : \mathrm{c}(F) \times F \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$
$mul_{\mathrm{c}(F),F}(x \mathbf{+}\hat{\mathbf{i}}\cdot\, y, z)$
$\qquad = mul_F(x, z) \mathbf{+}\hat{\mathbf{i}}\cdot\, mul_F(y, z)$

$mul_{\mathrm{i}(F),\mathrm{c}(F)} : \mathrm{i}(F) \times \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$
$mul_{\mathrm{i}(F),\mathrm{c}(F)}(\hat{\mathbf{i}}\cdot\, y, z \mathbf{+}\hat{\mathbf{i}}\cdot\, w)$
$\qquad = neg_F(mul_F(y, w)) \mathbf{+}\hat{\mathbf{i}}\cdot\, mul_F(y, z)$

$mul_{\mathrm{c}(F),\mathrm{i}(F)} : \mathrm{c}(F) \times \mathrm{i}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}\}$
$mul_{\mathrm{c}(F),\mathrm{i}(F)}(x \mathbf{+}\hat{\mathbf{i}}\cdot\, y, \hat{\mathbf{i}}\cdot\, w)$
$\qquad = neg_F(mul_F(y, w)) \mathbf{+}\hat{\mathbf{i}}\cdot\, mul_F(x, w)$

NOTE 2 – $mul_{\mathrm{c}(F)}$ is specified in clause 5.2.6.

$div_{\mathrm{i}(F)} : \mathrm{i}(F) \times \mathrm{i}(F) \rightarrow F \cup \{\mathbf{-0}, \mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$
$div_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot\, y, \hat{\mathbf{i}}\cdot\, w) = div_F(y, w)$

$div_{F,\mathrm{i}(F)} : F \times \mathrm{i}(F) \rightarrow \mathrm{i}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$
$div_{F,\mathrm{i}(F)}(x, \hat{\mathbf{i}}\cdot\, w) = \hat{\mathbf{i}}\cdot\, neg_F(div_F(x, w))$

$div_{\mathrm{i}(F),F} : \mathrm{i}(F) \times F \rightarrow \mathrm{i}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$
$div_{\mathrm{i}(F),F}(\hat{\mathbf{i}}\cdot\, y, z) = \hat{\mathbf{i}}\cdot\, div_F(y, z)$

$div_{F,\mathrm{c}(F)} : F \times \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$
$div_{F,\mathrm{c}(F)}(x, z \mathbf{+}\hat{\mathbf{i}}\cdot\, w)$
$\qquad = div_{\mathrm{c}(F)}(x \mathbf{+}\hat{\mathbf{i}}\cdot\, im_F(x), z \mathbf{+}\hat{\mathbf{i}}\cdot\, w)$

$div_{\mathrm{c}(F),F} : \mathrm{c}(F) \times F \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$
$div_{\mathrm{c}(F),F}(x \mathbf{+}\hat{\mathbf{i}}\cdot\, y, z)$
$\qquad = div_F(x, z) \mathbf{+}\hat{\mathbf{i}}\cdot\, div_F(y, z)$

$div_{\mathrm{i}(F),\mathrm{c}(F)} : \mathrm{i}(F) \times \mathrm{c}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$
$div_{\mathrm{i}(F),\mathrm{c}(F)}(\hat{\mathbf{i}}\cdot\, y, z \mathbf{+}\hat{\mathbf{i}}\cdot\, w)$
$\qquad = div_{\mathrm{c}(F)}(re_{\mathrm{i}(F)}(\hat{\mathbf{i}}\cdot\, y) \mathbf{+}\hat{\mathbf{i}}\cdot\, y, z \mathbf{+}\hat{\mathbf{i}}\cdot\, w)$

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 10967-3:2006

$div_{\mathrm{c}(F),\mathrm{i}(F)} : \mathrm{c}(F) \times \mathrm{i}(F) \rightarrow \mathrm{c}(F \cup \{\mathbf{-0}\}) \cup \{\mathbf{underflow}, \mathbf{overflow}, \mathbf{infinitary}, \mathbf{invalid}\}$

$div_{\mathrm{c}(F),\mathrm{i}(F)}(x \mathbf{+\hat{\imath}\cdot} y, \mathbf{\hat{\imath}\cdot} w)$
$\qquad = div_F(y, w) \mathbf{+\hat{\imath}\cdot} neg_F(div_F(x, w))$

NOTE 3 – $div_{\mathrm{c}(F)}$ is specified in clause 5.2.6.

5.2.5.6 Absolute value, phase and signum of complex floating point values

NOTES

1 This clause also includes $signum_F$.

2 abs_F is specified in part 1.

$abs_{\mathrm{i}(F)} : \mathrm{i}(F) \rightarrow F$

$abs_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y) \quad = abs_F(y)$

$abs_{\mathrm{c}(F)} : \mathrm{c}(F) \rightarrow F \cup \{\mathbf{underflow}, \mathbf{overflow}\}$

$abs_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y) \quad = hypot_F(x, y)$

$phase_F : F \rightarrow F$

$phase_F(x) \quad = arc_F(x, im_F(x))$

NOTE 3 – The arc_F (and similarly $arcu_F$) operation as specified in clause 5.3.8.15 of the *first edition* (issued in 2001) of part 2 has a minor flaw; it is not properly limited. See annex F for a corrected version. Implementations are recommended to follow this improvement, if possible, for arc_F, $arcu_F$, and all operations that reference those specifications.

$phase_{\mathrm{i}(F)} : \mathrm{i}(F) \rightarrow F$

$phase_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y) \quad = arc_F(re_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y), y)$

$phase_{\mathrm{c}(F)} : \mathrm{c}(F) \rightarrow F \cup \{\mathbf{underflow}\}$

$phase_{\mathrm{c}(F)}(x \mathbf{+\hat{\imath}\cdot} y)$
$\qquad = arc_F(x, y)$

$phaseu_F : F \times F \rightarrow F \cup \{\mathbf{invalid}\}$

$phaseu_F(u, x) \quad = arcu_F(u, x, im_F(x))$

$phaseu_{\mathrm{i}(F)} : F \times \mathrm{i}(F) \rightarrow F \cup \{\mathbf{invalid}\}$

$phaseu_{\mathrm{i}(F)}(u, \mathbf{\hat{\imath}\cdot} y)$
$\qquad = arcu_F(u, re_{\mathrm{i}(F)}(\mathbf{\hat{\imath}\cdot} y), y)$

$phaseu_{\mathrm{c}(F)} : F \times \mathrm{c}(F) \rightarrow F \cup \{\mathbf{underflow}, \mathbf{invalid}\}$

$phaseu_{\mathrm{c}(F)}(u, x \mathbf{+\hat{\imath}\cdot} y)$
$\qquad = arcu_F(u, x, y)$

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 10967-3:2006