ICS 35.160
L 62

中华人民共和国国家标准

GB/T 17966—2000
idt IEC 559:1989

微处理器系统的二进制浮点运算

Binary floating-point arithmetic for microprocessor systems

2000-01-03 发布　　2000-08-01 实施

国家质量技术监督局　发布

前　　言

本标准等同采用国际标准 IEC 559:1989《微处理机系统的二进制浮点运算》。

本标准是微处理机系统二进制浮点运算的标准,它包括的二进制浮点运算可由计算机软件、硬件以及软硬结合的方法来实现,本标准是计算机的基础标准。

本标准的附录 A 是提示的附录。

本标准由中华人民共和国信息产业部提出。

本标准由中国电子技术标准化研究所归口。

本标准起草单位:中国电子技术标准化研究所。

本标准主要起草人:高健。

IEC 前言

1) IEC(国际电工委员会)在技术问题上的正式决议和协议,是由对这些问题特别关切的国家委员会参加的技术委员会制定的,对所涉及的问题尽可能地代表了国际上的一致意见。

2) 这些决议或协议,以推荐标准的形式供国际上使用,并在此意义上为各国家委员会所认可。

3) 为了促进国际上的统一,IEC 希望各国家委员会在本国条件许可的情况下,采用 IEC 标准的文本作为其国家标准。IEC 标准与相应国家标准之间的差异,应尽可能在国家标准中指明。

IEC 序言

本国际标准由 IEC 的 TC 47“半导体设备”技术委员会 47B“微处理器系统”分委员会制定(本分委员会已由 ISO/IEC JTC 1 接管)。

IEC 出版的 559 第二版代替 1982 年发行的第一版。

本标准依据下列文件:

六个月的规则	投票报告
47B(CO)19	47B(CO)26

在上表指定的投票报告能找到采纳本标准的全部投票信息。

中华人民共和国国家标准

微处理器系统的二进制浮点运算

GB/T 17966—2000
idt IEC 559:1989

Binary floating-point arithmetic for microprocessor systems

1 范围

1.1 实现目标

其宗旨在于,无论是纯软件、纯硬件或软硬件组合的方法都能实现符合本标准的浮点系统。系统的程序员或系统用户可以知道是否符合本标准。要求软件支持才能符合本标准的硬件部分,离开软件便不能看作与本标准符合。

1.2 包含

本标准规定:

1) 基本和扩充的浮点数格式;
2) 加、减、乘、除、开平方、求余数以及比较运算;
3) 整数与浮点数之间的转换;
4) 不同浮点格式之间的转换;
5) 基本格式浮点数与十进制数串之间的转换;
6) 浮点异常及其处理,包括非数(NaN)的处理。

1.3 不包含

本标准不规定:

1) 十进制串和整数的格式;
2) NaN 符号和有效字段的解释;
3) 扩充格式二进制←→十进制之间的转换。

2 定义

本标准采用下列定义。

2.1 有偏阶码 biased exponent

阶码与选定常数(偏值)之和,所选常数使有偏阶码不出现负值。

2.2 二进制浮点数 binary floating-point number

由符号、带符号的阶码和有效数三部分表征的位串。如果它的数值值存在,则是它的有效数与 2 的阶码幂次的带符号的乘积。在本标准中,位串与它表示的数通常不加区别。

2.3 反规格化数 denormalized number

非零的浮点数,阶码有一保留值通常是格式的最小值,而且其显式或隐式有效首位等于零。

2.4 目的地 destination

二元或一元运算结果的位置。目的地由用户显式地指定或者由系统隐式地提供(例如,各过程的子表达式或自变量的中间结果)。一些语言,把中间结果放置在用户不能控制的目的地。本标准仍然按照目的地格式以及操作数的值来定义运算结果。

2.5 阶码 exponent

国家质量技术监督局 2000-01-03 批准　　2000-08-01 实施

二进制浮点数的组成部分，在确定浮点数所代表的数值时，它通常表示 2 的整数次幂。有时阶码也称作带符号阶码或无偏阶码。

2.6 小数 **fraction**

有效数中的一部分，它位于有效数隐含的二进制小数点右边。

2.7 方式 **mode**

用户可以设置、读出、保存和恢复的变量，用来控制后续算术运算的执行。默认方式是只要在程序或程序说明中无明显的矛盾语句，程序就可以假定是有效的方式。

应实现以下方式：

1) 舍入，以控制舍入误差的方向以及某些实现的舍入误差。

2) 舍入精度，降低结果的精度。实现者可以任意实现以下方式。

3) 禁止自陷/允许自陷，以处理异常情况。

2.8 非数 **not a number(NaN)**

不是一个数，用浮点格式编码的符号实体。**NaN** 有两种类型(见 6.2)。只要 **NaN** 作为操作数出现，信号 **NaN** 都发出无效操作异常信号(见 7.1)。静默 **NaN** 通过几乎每个算术运算进行传播而无需信号异常。

2.9 结果 **result**

递交给目的地的位串(通常表示数字)。

2.10 有效数 **significant**

二进制浮点数的组成部分，由二进制小数点左边的显式或隐式首位和小数点右边的小数部分所组成。

2.11 应 **shall**

指符合标准的实现是必须的。

2.12 宜 **should**

指极力推荐，以跟上标准的意向。虽然超出本标准范围之外的体系结构等方面的约束有时使这种推荐难于实现。

2.13 状态旗标 **status flag**

可取设置和清除两种状态的变量。用户可以对先前状态清除、拷贝或者恢复旗标。当为设置状态时，状态旗标可含有某些用户可能无法访问的系统相关的附加信息。作为某一方面的作用，本标准的操作可以设置部分下列旗标：不精确的结果、下溢、上溢、被零除以及无效运算。

2.14 用户 **user**

任何个人、硬件或程序(在其他标准已有规定)可以访问和控制本标准规定的程序设计环境的那些运算。

3 格式

本标准定义了四种浮点格式，分为基本格式和扩充格式两类。每一类又分为单精度和双精度两种。实现本标准的等级根据支持格式的组合来区分。

3.1 值集

本条仅涉及格式内可表示的数值值，不涉及以下各条所讨论对象的编码。选定格式的可表示值仅由以下三种整型参数来规定：

P——有效位数(精度)

$\mathbf{E}_{max}$＝最大阶码

$\mathbf{E}_{min}$＝最小阶码

表 1 列出每种格式的参数。各个格式都应提供形式为$(-1)^s 2^E (b_0 b_1 b_2 \cdots b_{p-1})$的数。

其中：s 是 0 或 1；

E 是 E_{min} 和 E_{max} 之间的任何整数(含 E_{min} 和 E_{max})；

b_i 是 0 或 1。

两种无穷+∞和−∞；

至少一个信号 NaN，并且至少一个静默 NaN。

表 1　格式的参数一览表

参　数	格　式			
	单精度	单精度扩充	双精度	双精度扩充
P	24	≥32	53	≥64
E_{max}	+127	≥+1023	+1023	≥+16383
E_{min}	−126	≤−1022	−1022	≤−16382
阶码偏数	+127	未规定	+1023	未规定
阶码宽度(位)	8	≥11	11	≥15
格式宽度(位)	32	≥43	64	≥79

以上所述可能导致值的冗余，比如：

$2^0(1.0)=2^1(0.1)=2^2(0.01)=\cdots$

然而，这些非零值的编码只在扩充格式下可能是冗余的(见 3.3)。形式为 $\pm 2^{E_{min}}(0.b_1b_2\cdots b_{p-1})$ 的非零值称为反规格化的。其他阶码也可以对 NaN、±∞、±0 和反规格化数进行编码。对于值为零的任何变数，符号位 s 提供额外的信息位。虽然所有格式都有区别地表示+0 或−0，但符号在一些情况下有意义，比如被零除，在其他情况下无意义。在本标准中，符号无关紧要时 0 和∞不带符号。

3.2　基本格式

单精度以及双精度格式的数都由三个域构成：

一位符号 s，

有偏阶码 e=E+偏值，

小数 $f=.b_1b_2\cdots b_{p-1}$。

无偏阶码 E 的范围应包括值 E_{min} 和 E_{max} 之间的每个整数(含 E_{min} 和 E_{max})，并且还有两个其他保留值：$E_{min}-1$ 用来对±0 以及反规格化数进行编码，$E_{max}+1$ 给±∞以及 NaN 进行编码。这些参数在表 1 中出现。每个非零数值值仅有一个编码。各部分在下面说明：

3.2.1　单精度

32 位单精度格式数 X 按图 1 所示划分。X 的值 v 是按照其组成部分导出，因此：

1) 如果 e=255，而且 f≠0，则不管 s 是什么，v 等于 NaN，

2) 如果 e=255，而且 f=0，则 $v=(-1)^s\infty$，

3) 如果 0<e<255，则 $v=(-1)^s2^{e-127}(1.f)$，

4) 如果 e=0，而且 f≠0，则 $v=(-1)^s2^{-126}(0.f)$(反规格化数)，

5) 如果 e=0，而且 f=0，则 $v=(-1)^s0$(零)。

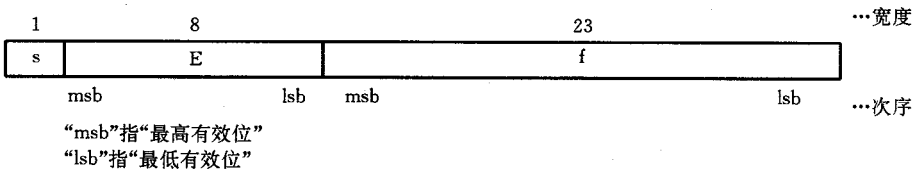

图 1　单精度格式

3.2.2 双精度

64 位双精度格式数 X 按图 2 所示划分。X 的 v 值是按照其组成部分导出，因此：

1) 如果 e=2047，而且 f≠0，则不管 s 是什么，v 等于 NaN，

2) 如果 e=2047，而且 f=0，则 $v=(-1)^s\infty$，

3) 如果 0<e<2047，则 $v=(-1)^s 2^{e-1023}(1.f)$，

4) 如果 e=0，而且同时 f≠0，则 $v=(-1)^s 2^{-1022}(0.f)$(反规格化数)，

5) 如果 e=0，而且 f=0，则 $v=(-1)^s 0$(零)。

图 2 双精度格式

3.3 扩充格式

单精度扩充和双精度扩充格式按依赖于实现的方法进行编码但受 3.1 中表 1 值的限制。本标准允许出现编码的冗余值，但这种冗余对用户透明，即按下面来理解：一个实现或者应对每个非零值唯一地编码，或者对非零值的冗余编码不加区分。一个实现也可以保留一些位串，用于超出本标准范围的各种用途；这种保留的位串作为操作数出现时，运算的结果本标准不作规定。

本标准的实现并不要求提供(并且用户不宜假定)单精度扩充格式范围比双精度扩充格式范围大。

3.4 格式组合

符合本标准的所有实现都应支持单精度格式。实现宜支持所支持的最宽基本格式相应的扩充格式，而不必支持任何其他扩充格式[1)]。

4 舍入

舍入时认为取数是无限精确的，如有必要，修正该数使之适合于目的地的格式，同时发出不精确的异常信号(见 7.5)。除二进制←→十进制转换外(5.6 规定其最低条件)，应完成第 5 章中规定的每种运算，同时在不受限制的范围下，先产生正确的无限精度的中间结果，然后按本章的方式对结果舍入。

除比较大小和求余数外，舍入方式影响所有的算术运算。舍入方式可以影响零和的符号(见 6.3)，同时影响所发出上溢(见 7.3)和下溢(见 7.4)信号时所超出的阈值。

4.1 舍入到最近值

本标准的实现应达到最近值舍入，并作为舍入的默认方式。在这个方式下，应递交最近于无穷精度结果的表示值；如这两个表示值同等接近，则应递交最低有效位等于零的一个。但是，数值不小于 $2^{E_{max}}(2-2^{-P})$的无穷精度结果，应舍入为符号不改变的∞；除舍入的精度方式(见 4.3)超越目的地格式外，其中的 E_{max} 和 P 由目的地格式(第 3 章)决定。

4.2 有向舍入

实现应给用户提供三种可选择的有向舍入方式：向+∞的舍入、向−∞的舍入以及向 0 舍入。

向+∞舍入时，其结果应是最接近于且不小于无限精确结果的格式值(可能是+∞)。向−∞舍入时，其结果应是最接近于且不大于无限精确结果的格式值(可能是−∞)。

向 0 舍入时，其结果应是最接近且数量上不大于无限精确结果的格式值。

4.3 舍入精度

通常，指按目的地格式的精度舍入结果。然而，某些系统仅对双精度或扩充精度格式目的地递交结果。对于这类系统，用户可以是高级语言编译器，应能规定把以双精度格式或具有更宽阶码范围扩充格

1) 只有向上兼容和速度是重要问题时，支持双精度扩充格式的系统也支持单精度扩充格式。

式存放的结果，舍入成单精度[1)]。与此类似，仅向双精度扩充格式目的地递交结果的系统，应允许用户规定将结果舍入成单精度或双精度。注意：为了符合 4.1 的规定，结果舍入误差不允许多于 1 个。

5 运算

符合本标准的所有实现都应提供加、减、乘、除、开平方、求余数，浮点格式舍入成整数，不同浮点格式间的转换，浮点与整数格式间的转换，二进制↔十进制转换以及比较运算。不改变格式的拷贝，是否作为一种运算，是实现的选项。除了二进制↔十进制转换外，完成每一种运算时在不受限制的范围中，先产生一个具有无限精度的中间结果，然后修正这种中间结果使之适合目的地的格式(见第 4 和第 7 章)。第 6 章将以下规定扩充至包括±0、±∞以及 NaN；第 7 章枚举异常操作数和异常结果导致的异常。

5.1 算术运算

实现对格式相同的任何两个操作数和所支持的格式的每一操作数都应提供加、减、乘、除和求余数运算；实现也应对不同格式操作数提供的运算。目的地格式(不管 4.3 的舍入精度控制如何)应至少与较宽的操作数格式一样宽。所有结果都应按第 4 章的规定舍入。

当 $y \neq 0$ 时，根据 $r = x - y \times n$ 关系式定义余数 $r = x$ REM y，与舍入方式无关，其中 n 是最接近于 x/y 精确值的整数；只要 $|n - x/y| = 1/2$，n 就是偶数。因此，余数总是精确的。如果 $r = 0$，它的符号应是 x 的符号。精度控制(见 4.3)应不适用于求余数操作。

5.2 开平方

对所有支持的格式都应提供开平方运算，对于所有非负的操作数，结果都是确定的并有一个正号，但$\sqrt{-0}$应为 −0 除外。目的地格式应至少与操作数格式一样宽。运算结果应按第 4 章的规定舍入。

5.3 浮点格式转换

在所有支持的格式之间都应能进行浮点数转换。如果转换成较低的精度时，其结果应按第 4 章的规定舍入。转换成较高的精度应是精确的，没有例外。

5.4 浮点与整数间的转换

所有支持的浮点格式与所有支持的整数格式之间应能进行转换。转换到整数应受第 4 章规定的舍入影响。浮点整数和整数格式间的转换应是精确的，7.1 中规定的异常情况除外。

5.5 浮点数舍入为整数值

浮点数应能舍入为同一格式的整数值浮点数。舍入应按第 4 章的规定，条件是当向最接近值进行舍入时，如果未舍入的操作数与舍入的结果之差恰为 1/2，则舍入结果是偶数。

5.6 二进制↔十进制转换

对于在表 2 规定的范围之内的所有数，应提供至少一种格式的十进制数串与所有支持的基本格式的二进制浮点数之间转换。表 2 和表 3 中的整数 M 和 N 是具有值为$\pm M \times 10^{\pm N}$的十进制串。输入时，为使 N 达到最小，应在 M(不超过表 2 的限度)的末尾增加或删除零。当目的地是十进制串时，为了舍入，最好按格式规定设置该串的最低有效位。

如果整数 M 不属于表 2 和 3 规定的范围，即，当单精度的 $M > 10^9$，双精度的 $M > 10^{17}$ 时，实现者可以故意在单精度的第 9 位和双精度的第 17 位，对后边的所有有效数字变换为其他十进制数字，典型的数字是 0。

对于不超过表 3 规定范围的操作数，转换时应按第 4 章规定正确地舍入。另外，为舍入到最接近值，只要不出现阶的上溢/下溢，转换结果的误差不应超过第 4 章舍入规定引起的误差在目的地最小有效数

1) 舍入精度的控制，本意是允许系统在运算中一般具有双精度或扩充格式的目的地，在没有上/下溢出时，系统精度为带单精度格式和双精度格式的目的地。实现不宜给出结合双精度或扩充运算产生单精度结果的操作，仅对一种舍入，也不宜给出结合双精度扩充运算产生双精度结果的操作。

的 0.47 单位。有向舍入方式的误差应有正确的符号同时在最后一位不超过 1.47 单位。

转换应是单调的。也就是，值递增的二进制浮点数转换成十进制数串时，其值不应减少；而值递增的十进制数串转换成二进制浮点数时，其值也不应减少。

当舍入到最接近值时，只要十进制数达到表 2 规定的最大精度，即单精度 9 位数字和双精度 17 位数字，二进制到十进制又返回到二进制的转换应是恒等的[1)]。

如果十进制到二进制的转换上溢/下溢，则按第 7 章的规定处理。在二进制转换成十进制的过程中遇到上溢/下溢、NaN 和无穷，则最好通过适当的串指示给用户。

为了避免冲突，二进制↔十进制转换过程应给出一样的结果，而不管转换是否是在语言转换(翻译、编译或汇编)期间或程序执行(运行和交互输入/输出)期间进行。

表 2　十进制转换范围

格　式	十进制到二进制		二进制到十进制	
	最大 **M**	最大 **N**	最大 **M**	最大 **N**
单精度	10^9-1	99	10^9-1	53
双精度	$10^{17}-1$	999	$10^{17}-1$	340

表 3　正确舍入的十进制转换范围

格　式	十进制到二进制		二进制到十进制	
	最大 **M**	最大 **N**	最大 **M**	最大 **N**
单精度	10^9-1	13	10^9-1	13
双精度	$10^{17}-1$	27	$10^{17}-1$	27

5.7　比较

在所有支持的格式中，即使操作数的格式不同，对浮点数都应能进行比较。比较是精确的，不上溢也不下溢。四种互斥的关系可能是：“小于”、“等于”、“大于”、“无序”。当至少一个操作数是 NaN 时，最后一种情况出现。每个 NaN 应与任何数(包括它自己)进行比较，但都“无序”。零的符号在比较时忽略(于是+0=−0)。

比较的结果应按下面这两种方式之一来递交：可以是条件代码，表示上面四种关系之一，也可以是命名所需特定比较关系谓词的真假响应。除真假响应外，当比较“无序”操作数时，使用包括“<”或“>”的但不要“?”的谓词(这里的符号“?”表示“无序”)，应按在表 4 的最后一列指出的来发出无效运算异常信号(见 7.1)。

表 4 列出二十六个功能不同的常用的谓词，在第一列中使用三种记法命名谓词：专设的、类似 FORTRAN 语言的和数学的。表中示出从四个条件代码中如何得到它们，并指出当关系“无序”时，哪些谓词导致无效运算异常。T 和 F 项指明各对应关系成立时谓词是真或假。

表 4　谓词和关系

谓　词			关　系				异　常
专设的	FORTRAN	数学的	大　于	小　于	等　于	无　序	无序是否无效
=	.EQ.	=	F	F	T	F	否
? <>	.NE.	≠	T	T	F	T	否

1) 转换所规定的特性暗含着差错界线，它取决于格式(单或双精度)及包含的十进制数字的数；上面所述的 0.47 仅是最坏情况下的界线。对于差错界线以及对扩充格式有效转换算法研究的详细讨论见加利福尼亚伯克利大学 Dissertation 著的“Accurate Yet Economical Binary-Decimal Conversions”。

表 4(完)

谓　词			关　系				异　常
专设的	FORTRAN	数学的	大　于	小　于	等　于	无　序	无序是否无效
>	.GT.	>	T	F	F	F	是
>=	.GE.	≥	T	F	T	F	是
<	.LT.	<	F	T	F	F	是
<=	.LE.	≤	F	T	T	F	是
?	无序		F	F	F	T	否
<>	.LG.		T	T	F	F	是
<=>	.LEG.		T	T	T	F	是
? >	.UG.		T	F	F	T	否
? >=	.UGE.		T	F	T	T	否
? <	.UL.		F	T	F	T	否
? <=	.ULE.		F	T	T	T	否
? =	.UE.		F	F	T	T	否
NOT(>)			F	T	T	T	是
NOT(>=)			F	T	F	T	是
NOT(<)			T	F	T	T	是
NOT(<=)			T	F	F	T	是
NOT(?)			T	T	T	F	否
NOT(<>)			F	F	T	T	是
NOT(<=>)			F	F	F	T	是
NOT(? >)			F	T	T	F	否
NOT(? >=)			F	T	F	F	否
NOT(? <)			T	F	T	F	否
NOT(? <=)			T	F	F	F	否
NOT(? =)			T	T	F	F	否

注意谓词配对出现,每一个都在逻辑上否定另一个;应用前缀比如“NOT”来否定表 4 中的谓词,使关联项的真假判断相反,但表中最后一列各项不变[1)]。

实现给出的谓词应给出表 4 中的前 6 个谓词,同时给出第 7 个,以及逻辑上否定谓词的办法。

6　无穷、NaN 和带符号的零

6.1　无穷运算

无穷运算应作为任意大操作数实数运算的取极限来构造,其前提是极限存在。无穷应按仿射意见解

1) 谓词的逻辑否定有两种方式,一种是加前缀“非”,另一种是反向相关运算。比如:(X=Y)的逻辑否定可以写成非(X=Y)或(X? <>Y),两个表达式功能上都等同于(X≠Y)。但是,对于另一些谓词这种等同并不存在。比如(X<Y)的逻辑否定恰好是(X<Y),反相谓词(X? >=Y)的不同之处是,当 X 和 Y 是“无序”时,不发出无效操作异常信号。

释，即$-\infty<$每个有穷数$<+\infty$。

除在7.1对∞规定的无效操作外，∞的运算总是精确的，因而没有异常信号发出。仅在下列情况下才发出与∞有关的异常信号。

1) ∞由有限操作数上溢(见7.3)或被零除(7.2)来产生，且对应自陷被禁用；

2) ∞是无效操作数(7.1)。

6.2 NaN的运算

所有运算都应支持两种不同类型的NaN，即信号的和静默的。信号NaN给不在本标准范围的非初始化变量以及增强的类似运算(比如复数仿射无穷和极宽的范围)提供值。静默NaN根据实现者的自定的方式，给出从无效或不可利用数据及结果中承袭的追溯诊断信息。诊断信息在传播时要求NaN包含的信息通过运算操作和浮点格式转换加以保护。

信号NaN应是保留操作数，它对第5章列出的每种运算都发出无效操作异常信号(见7.1)。不改变格式的拷贝信号NaN是否发出无效运算异常信号，这由实现者选择。

如果没有自陷出现而又要递交浮点结果，则包括信号NaN或无效运算(见7.1)的每种运算都应递交一个静默NaN作为它的结果。

包括一个或两个输入NaN，但都不是信号NaN的每种运算不应发出异常信号，但如果递交浮点结果，则应递交一个静默NaN作为它的结果，它宜是输入NaN之一。注意格式转换时不可能递交同一NaN。静默NaN的影响有类似于信号NaN，对运算不递交浮点结果；这些运算：即对某一格式的比较和转换在5.4、5.6、5.7和7.1中讨论。

6.3 符号位

本标准不说明NaN的符号。而积和商的符号是操作数符号的“异或”；和的符号至多不同于一个加数的符号，差x—y可以认为是x+(—y)。即使操作数或结果是零或无穷，这些规则应适用。

当两个操作数的和(或两个符号相同的操作数之差)恰好是零时，除向$-\infty$舍入时则符号为“—”外，不论用哪种舍入方式，和的符号(或差的符号)均为“+”。然而，即使x是零，x+x=x—(—x)保持与x相同的符号。

除—0的平方根是—0外，所有有效平方根的符号都应为正号。

7 异常

检测后应发出的异常信号有五种类型。信号需设置状态标志、采取自陷，或两者都需。各异常在用户控制下应与一自陷关联，这在第8章规定。异常的默认响应应不带自陷进行。在自陷或非自陷情况下，本标准规定要递交的结果。在某些情况下，如果自陷是允许的，则结果不同。

对于每类型的异常，实现应提供状态标志，当没有相应自陷存在时，应将它设置在相应异常的每次出现上。仅在用户请求时，它才应重新置位。用户应能分别测试与变更状态标志，进而应能在同一时间保存和恢复所有五类型的异常。

能重合的异常仅为上溢不精确与下溢不精确。

7.1 无效运算

如果执行运算的操作数无效，则应发出无效操作异常信号。当没有自陷的异常出现时，只要提供目的地有浮点格式，结果就应是静止NaN。无效运算是：

1) 对信号NaN的任何运算(见6.2)；

2) 加或减：像$(+\infty)+(-\infty)$这样的无穷的数量减；

3) 乘：$0\times\infty$；

4) 除：0/0或∞/∞；

5) 余数：x REM y；其中y是零或x是无穷；

6) 操作数小于零的开平方；

7) 当上溢、无穷或 NaN 排除这一格式可靠表示但不能就此发出其他的信号时，二进制转换成整数或十进制格式；

8) 操作数是“无序”(见 5.7，表 4)时，通过涉及谓词“<”或“>”但没有“?”的比较。

7.2 被零除

如果除数是零而被除数是有限非零数，则应发出被零除异常信号。当无自陷出现时，结果应是带正确符号的∞(见 6.3)。

7.3 上溢

凡目的地格式的最大有限数在大小上为舍入浮点结果(第 4 章)阶码范围无界所超出时，都发出上溢异常信号。当没有自陷出现时，应由舍入方式和中间结果的符号按下列方式来确定结果：

1) 最接近舍入将所有上溢进位成为带中间结果符号的∞；

2) 向 0 舍入将所有上溢进位成为带中间结果符号的格式的最大有限数；

3) 向−∞舍入将正上溢进位成为格式的最大有限数，同时将负上溢进位成为−∞；

4) 向+∞舍入将负上溢进位成为格式的最大的负有限数，同时将正溢进位成为+∞。

除转换外，所有操作的自陷上溢都应通过无穷精确的结果除以 2^a 和舍入得到的结果，递交给自陷的处理器。偏差调节 a 在单精度是 192，在双精度是 1536，扩充格式是 $3\times2^{n-2}$，其中，n 是阶码域的位数。由二进制浮点格式转换产生的自陷上溢出应给自陷处理者递交一个结果，这一结果按此格式或更宽的格式——可能带有阶码偏差调节——但要舍入到目的地精度。十进制到二进制转换的自陷上溢出也应给处理者递交一个结果，它按照支持的最宽的格式——可能带有阶码偏关设列关能文地设票见致形关设列关能作下地

序。当发出自陷禁止的异常信号时，应按第7章规定的方式来处理它。当发出自陷允许的异常信号时，出现异常的程序应暂停执行，激活由用户先前定的自陷处理程序，同时，如果按第7章规定，则应把结果递交给它。

8.1 自陷处理程序

自陷处理程序最好有子例行程序的能力，它能返回一个有用值代替异常运算结果；除通过自陷处理程序提递交外，这个结果未定义。类似地，以其所允许的关联自陷发出异常信号的标志若不由自陷处理程序来置位或复位，该标志也未定义。

当系统自陷时，自陷处理程序最好能确定：

1) 这一运算发生的异常；

2) 所执行的运算种类；

3) 目的地格式；

4) 在上溢、下溢、不精确异常中，已正确舍入的结果包括可能不适于目的地格式的信息；

5) 在无效操作数和被零除异常中的操作数值。

8.2 优先

如果允许，上溢和下溢自陷获得优先于独立的不精确自陷。

附 录 A
（提示的附录）
推荐的函数和谓词

以下是推荐的函数和谓词，它有助于跨越不同系统（也许完成运算的方式差异很大）的程序可移植性。一般地描述了它们；即操作数和结果类型是操作数固有的。要求显式分类型的语言要有相应的函数和谓词族。

下面的一些函数，像不改变的格式拷贝运算 y:=x 一样，在实现者的选项上，可看作非算术运算，该运算对信号 NaN 不发出无效运算异常信号，所研究的函数是 1)、2)、6)和 7)。

1) copysign(x,y)返回带有 y 的符号 x。于是有 abs(x)=copysign(x,1.0)，即使 x 是 NaN。

2) −x 是指以其相反符号拷贝成的 x，而不是 0−x；当 x 是±0 或 NaN 时，这种是区别密切关系的。因此，使用符号位去区分静默 NaN 和信号 NaN 是错误的。

3) scalb(y,N)对整数 N 返回 $y\times 2^N$；而无需计算 2^N。

4) logb(x)返回 x 的无偏阶码（x 的格式中的带符号整数），但 logb(0)是−∞，logb(∞)是+∞，与 logb(NaN)是 NaN 除外，并发出以零除的异常信号。当 x 为正且有限时，表达式 scalb(x,−logb(x))严格介于 0 与 2 之间；仅当 x 反正规化时才小于 1。

5) nextafter(x,y)沿向 y 方向返回下一个 x 的下一个可表示的相邻值。出现以下特殊情况：若 x=y，则结果是 x，而不发出任何异常信号；否则，若 x 或 y 都不是静默 NaN，则结果是输入 NaN 中的一个或别一个。当 x 有限而 nextafter(x,y)无穷时，发出上溢信号；当 nextafter(x,y)严格介于$\pm 2^{E_{min}}$间时，发出下溢信号；在这两种情况下，都要发出不精确信号。

6) 若−∞<x<+∞，则 finite(x)返回值 TRUE，否则返回值 FALSE。

7) 若 x 是 NaN，isnan(x)或与此等价的 x≠x 返回值 TRUE，否则返回值 FALSE。

8) 仅当 x<y 或 x>y 时，x<>y 才是 TRUE，它与 x=y 不同，它意指 NOT(x=y)（表 4）。

9) 如果 x 对 y 是无序的，则 unordered(x,y)或 x? y 返回值是 TRUE，否则返回 FALSE（表 4）。

10) class(x)分出 x 属于下列十类中哪一类：信号 NaN、静默 NaN、−∞、负的规格化非零、负的反规格化的−0、+0、正的反规格化的非零、+∞、正的规格化的非零、+∞。该函数即使不是对于信号 NaN 的，该函数也绝不是异常的。