Submitter: Fred Tydeman and Jim Thomas (USA)

Submission Date: 2009-05-04

Previous version of paper: N1303, N1317, N1352, N1378

Related WG14 documents: N1151, N1171

Related WG21 documents: N2798

Subject: xxx_TRUE_MIN macros for <float.h>

This is a rewrite based upon feedback from the committee.

Changes to C1x

Add new bullets to **5.2.4.2.2 Characteristics of floating types <float.h>**

[bullet after FLT_EVAL_METHOD] the presence or absence of subnormal numbers is characterized by the values **FLT_HAS_SUBNORM**, **DBL_HAS_SUBNORM**, and **LDBL_HAS_SUBNORM**:

```
-1      indeterminable (footnote A)
0       absent (footnote B) (type does not allow subnormal numbers)
+1      present (type does allow subnormal numbers)
```

A) Characterization as indeterminable is intended if floating-point operations do not consistently interpret subnormal representations as zero, nor as nonzero.

B) Characterization as absent is intended if no floating-point operations produce subnormal results from non-subnormal inputs, even if the type format includes representations of subnormal numbers.

[bullet after FLT_MIN] minimum positive floating-point number (footnote C)

```
FLT_TRUE_MIN        1E-37
DBL_TRUE_MIN        1E-37
LDBL_TRUE_MIN       1E-37
```

C) If the presence or absence of subnormal numbers is indeterminable, then the value is intended to be a positive number no greater than the minimum normalized positive number for the type.

[paragraph 15, example 2] Remove "normalized" from just before IEC 60559.

Add

```
FLT_TRUE_MIN        1.40129846E-45 // decimal constant
FLT_TRUE_MIN        0X1P-149F // hex constant
FLT_HAS_SUBNORM      +1
DBL_TRUE_MIN        4.9406564584124654E-324 // decimal constant
DBL_TRUE_MIN        0X1P-1074 // hex constant
DBL_HAS_SUBNORM      +1
```

after **FLT_MIN** and **DBL_MIN**.

Words for Rationale:

[add to 5.2.4.2.2 section] Applications that depend on support for subnormal numbers (i.e., gradual underflow) can check, at translation time, if xxx_HAS_SUBNORM has the value +1, where

xxx is FLT, DBL, or LDBL. Applications that depend on subnormal numbers not arising in computations can check if xxx_HAS_SUBNORM has the value 0.

Today, most processors treat subnormal numbers according to IEC 60559, supporting xxx_HAS_SUBNORM = +1; or, they flush subnormal results (and perhaps also subnormal operands) to zero, supporting xxx_HAS_SUBNORM = 0. For processors that provide a control mode to do one or the other, the determination of xxx_HAS_SUBNORM depends on the startup setting of the control mode, because the standard does not provide means to change the control mode.

Implementations whose floating-point operations are not consistent in their treatment of subnormal representations should define xxx_HAS_SUBNORMAL to -1. Prior to IEEE 754-1985, processors treated subnormal representations in a great variety of ways, sometimes with bizarre and problematic results. See http://expertvoices.nsdl.org/cornell-cs322/2008/03/26/the-old-man-of-floating-point-and-subnormal-numbers/ and http://www.quadibloc.com/comp/cp0201.htm for some examples.

The values of the smallest subnormal floating-point numbers (if supported) are typically, but not always, FLT_MIN*FLT_EPSILON, DBL_MIN*DBL_EPSILON, LDBL_MIN*LDBL_EPSILON. The Motorola 68881 supports an 80-bit format whose minimum subnormal is a factor of 2 smaller than the formula would indicate, which was valid in IEEE 754-1985 (equivalent to IEC 60559), but not in IEEE 754-2008.