Document: WG14/N1213
Date: 2007/03/26
Project: TR 24732
Authors: Jim Thomas, Rich Peterson
Reply to: Rich Peterson <Rich.Peterson@hp.com>

Subject: limiting _Decimal32 functionality

This paper proposes changes to the _Decimal32 specification in WG14/N1201 (TR 24732 draft of 2006/11/10) to align with 754R's intended use for it.

Background: 754R specifies its 32-bit decimal format as a "storage format", which means it's intended for efficient storage, not computation. The only operations 754R requires for storage formats are conversions to and from other supported formats in the same radix. 754R does not disallow direct computation on operands in storage formats but recommends that languages permitting them perform such computations in wider formats. The 754R committee regarded the range and precision of storage formats to be too narrow, hence error prone, for general use.

The TR specification does permit arithmetic on _Decimal32 operands. This is allowed by 754R and seems reasonable because introducing a fundamentally different kind of limited floating type into C would be awkward at best. However, the TR specification unnecessarily runs counter to the 754R recommendation in two ways:

Problem 1: In section 5, page 7, in the evaluation methods represented by DEC_EVAL_METHODS, the method of evaluating all operations and constants just to the range and precision of the type, which 754R deprecates for implementations supporting the 32-bit decimal format, is presented on equal footing with the wide evaluation methods recommended by 754R. Also, allowing other implementation-defined or indeterminable evaluation is undesirable in a specification designed to support 754R decimal floating point exclusively.

Suggested TR changes for Problem 1:

In section 5, page 7, in TR-proposed 5.2.4.2.2a [#2], delete the bullets for DEC_EVAL_METHOD equal to -1 and 0 and delete the sentence following the bullets.

In section 9.3, page 20, remove "DECIMAL_EVAL_METHOD equals 0, _Decimal32_t and _Decimal64_t are _Decimal32 and _Decimal64, respectively; if" and also remove "; and for other values of DEC_EVAL_METHOD, they are otherwise implementation-defined".

Problem 2: In section 9.3, beginning on page 21, the specification for <math.h> encourages _Decimal32 computation by providing _Decimal32 versions of its functions .In section 9.8, page 33, <tgmath.h> encourages _Decimal32 computation by invoking _Decimal32 functions for type-generic function calls with _Decimal32 arguments.

Suggested TR changes for Problem 2:

In section 9.3, page 19, paragraph 2, delete "d32,".

In section 9.3, page 21 [#7], delete FP_FAST_FMAD32.

In section 9.3, pages 21-26, delete prototypes for all functions suffixed with "d32".

In section 9.8, page 33, replace the second and third bullets with the one bullet "Otherwise, if any argument has type _Decimal32 or _Decimal64, the type determined is _Decimal64."

This proposal does not suggest removing other library support for _Decimal32 which might be helpful for non-computational processing of_Decimal32 data.