

**WG14 N2199**  
**Meeting notes**

**C Floating Point Study Group Teleconference**

2018-01-09

9 AM PST / 12 PM EST

**Attendees:** Rajan, Jim, Ian, Fred, Mike, David H, Clark,

**New agenda items:**

DR16 (Clark)  
Fred email DECIMAL\_DIG  
Jim's email about TS update  
Fred's email regarding roundToEven

Ian: Retiring at end of May this year

Ian: POWER9 will have IEEE quad precision

**Last meeting action items:**

Ian: See if there is an incompatibility between C and C++ for constants being evaluated to a wider format (Ex. FLT\_EVAL\_METHOD affects constants in C++, and wider return values) - Keep open (Hubert: Not defined and left up to C)

Jim: tgm<sub>math</sub>\_for\_narrowing\_functions (DR13, part 3): Need to bring in functions that cover three arguments. - Done.

Jim: tgm<sub>math</sub>\_for\_narrowing\_functions (DR13, part 3): Rewording for the f32mul example. - Done.

Jim: Create a new paper along the lines of tgm<sub>math</sub>\_for\_narrowing\_functions proposing a new TC for CFP DR13. - Done.

Jim: Write a DR addressing WG14 reflector message 14885 using the email sent by Jim on 2017/11/13. - Done.

Fred: Look through the new functions and see what is missing in Annex F. - Not done.

Fred: Follow up with inconsistencies in Infinities references in function descriptions in our TS via email. - Done.

Fred: Look into 754:2018 4.3.1 (roundTiesToEven) applies to us for string conversion (printf for example). - Done.

Jim: Update the binding table in parts 1 and 2 to handle the new IEEE-754:2018 functions when published. - Keep open.

David H: Look into the consistency with our TS and IEEE-754:2018 for 5.3.2, 9.2, 9.2.1, 9.2.2 and 9.4. - Done.

Jim: Update activities list ([http://wiki.edg.com/pub/CFP/WebHome/in\\_flight-20171004.pdf](http://wiki.edg.com/pub/CFP/WebHome/in_flight-20171004.pdf)) with new status (LaTeX integration) and items (consistency with 754:2018 draft, special cases in Annex F). - Done.

**New action items:**

exp10m1: Look at exp10m1 difference between the TS and 754 in more detail.

pow: Add a note to F10.4.4 pow to say it is the same as IEEE-754.

reduc\_sumprod: "computed sum" -> "computed dot product" for clarity.

Jim: Add preferred exponents functions list to part 4.

Jim: Get a N# and post the new TC for DR13 to WG14.

Jim: Create a new DR for arguments for comparison macros

([http://wiki.edg.com/pub/CFP/WebHome/DR\\_for\\_incommensurate\\_arguments\\_for\\_comparison\\_macros.pdf](http://wiki.edg.com/pub/CFP/WebHome/DR_for_incommensurate_arguments_for_comparison_macros.pdf))

Fred: Draft a note for roundTiesToEven for the exceptional case of two odd values.

Jim: Draft a proposal for CR\_DECIMAL\_DIG corrections (with the removal of DECIMAL\_DIG) and updating the footnote (F.5).

Jim: DR15: Make the Suggested TC the Proposed TC.

Jim: Re-update the activities list from results from today.

**Issues:**

Does CR\_DECIMAL\_DIG have the same issues as DECIMAL\_DIG?

Why does the cbrt macro (DR16) have the parameters inside the generic selection?

**Next Meetings:**

February 20th, 2018, 12:00 EST, 9:00 PST

Same teleconference number.

**Discussion:**

IEEE 754 revision:

Adding payload functions found a lot of issues. Being resolved.

The existing 2008 standard expires at the end of 2018. Looking into extending the lifetime until the new one can take over.

C++ liaison:

Nothing.

C DR16:

Clark: Concerns: A lot of effort in improving an example. Also cascading error.

Jim: The change is not to the example, it is a TS description of what needs to be changed to support tgmth with this if it followed what the example did.

Clark: If the compiler knows that a math library function is being called, the implementation should honor the static rounding mode.

Jim: It's a part of the semantics of the macro. If you are using the macro (like the tgmth.h cbrt macro), this describes what you would expect.

Clark: Why are you trying to describe the implementation?

Jim: Since the C standard shows how it can be implemented for tgmth. This is a note to show what needs to be done to make it work. As it is it would not work the way it is (i.e. would not support static rounding modes).

Clark: History: If a library function was called indirectly it is not required to support static rounding modes. This was changed to the macro expansion text which is where you went off the rails.

Jim: If macro expansion has been disabled, the modes are not supported.

Jim: We do say explicitly that if you do suppress macro expansion you will not get static rounding.

Clark: The original example in the standard was carefully crafted to not have the argument list being listed multiple times.

Jim: Don't know why that was done. May have been a reason. We can put that aside.

Clark: Was the parameter list moved inside to allow further macro expansion?

Jim: We can look into why that was done.

Clark: Moving the argument list outside the selection and knowing you are doing what you are doing on purpose is a step in the right direction. Just keep me in the loop (send an email) about things that touch generic selection.

If the original motivation was to handle indirect calls, allowing a way to force not using static rounding modes is not good and this is the wrong way to go.

Email Issues:

Consistency between CFP TS and IEEE-754:2008 (David's email on 2018/01/08):

tanpi: Likely implementations present now.

David: IEEE didn't add it due to strange formats (not the zero and inf).

atan2pi: Jim: Domain errors is historical. Should be consistent otherwise.

expl0m1: David: Seems to be a mistake in the TS.

Jim: The difference in the value that is returned is too small to affect the result. i.e You can't get a

small and inexact result.

\*Look at exp10m1 issues in more detail.

Fred: Currently says range error if finite x is too large, but it should be positive finite x.

Jim: It is already a defect right? Can raise it via email if needed.

pow: David: We should translate the IEEE spec into the C spec as close as possible to make it easier to compare.

Jim: But comparisons between C versions would then change.

David: We could have a statement saying it is the same as 754, just in a different order.

\*Add a note to F10.4.4 pow to say it is the same as IEEE-754.

powr: \*Look into the difference between powr in the TS vs 754 in more detail.

Fred: We do list it as an error in part 4.

Remove the action item.

reduc\_sumprod: Jim: I don't think we use the term dot product anywhere.

David: We do. In 7.12.13b4.

\*reduc\_sumprod: "computed sum" -> "computed dot product" for clarity.

Preferred exponents: Missing entries for rsqrt, compound, rootn, pown, powr for preferred quantum exponents.

David: These are new functions (not in C) so they should not be in Part 2, but should be in Part 4.

\*Jim: We just missed them in part 4 I believe.

tgmath for narrowing functions (2017/11/18):

Jim: Can remove some extra wording ("determined", "real type") since it is already defined and determined.

New TC for DR13 (2018/01/02): Has all the changes.

\*AI: Jim: Get a document number and send this to WG14.

DR addressing arguments for comparison macros (2018/01/02):

Create a new DR for this.

Inconsistencies in specification of infinities (2018/01/08 email titled "fadd, fsub and domain errors"):

Fred: Not too hard make the changes.

Listing functions that list infinities and NaNs. There are quite a few that mention them. It is not just in Annex F.

nextafter: Jim: The infinite there is helpful since not representable may be confusing. Infinite is there since it is helpful.

The last part (754 vs 18661) is for another action item. Will follow up via email.

Jim: There is no attempt to cover all inf and nan cases.

Fred: If we keep the existing mentioning of inf and nan, we should add fadd and fsub or mention that domain errors occur for inf arguments with opposite sign for the TS.

Jim: If an implementation has unsigned inf, is a domain error allowed?

Fred: For fsub, yes, but not for fadd.

Fred to send an email if this proposal is wanted.

roundTiesToEven (2018/01/08, 09):

Fred: 754 requires taking the larger magnitude one if both are odd.

Our doc follows printf for rounding.

We should add this one example to show the unusual case.

Jim: Is a note OK?

Fred: Fine.

DECIMAL\_DIG (2017/11/19):

Fred: Without DECIMAL\_DIG, the other definitions need something (CR\_DECIMAL\_DIG).

Jim: We can have the formula for DECIMAL\_DIG + 3 to replace DECIMAL\_DIG.

It should support extended, interchange and non-arithmetic.

In part 1, we talk about supported formats, so CR\_DECIMAL\_DIG should be fine (vs the issue with DECIMAL\_DIG). In part 3 the other formats should have it working automatically.

\*Issue: Does CR\_DECIMAL\_DIG have the same issues as DECIMAL\_DIG.

\*Jim: Write up a proposal on changes based on this email.

This should not be a part of DR501 since it is for Part 1, not the C standard.

DR15 (2018/01/06):

Will make the change as editorial but keep the suggested TC as the proposed TC.

Binding for IEEE 754-2018:

Payload: We will have to change.

The rest seem fine (will discuss augmented arithmetic after with Mike and David present).

C2X integration:

Jim: Any guidelines for using git?

Rajan: Jens said anyone can create branches. We should get the N document that is submitted for FDIS.

Some committee members prefer the full document, not just the diff list.

Activities:

Defer to next meeting.

Other:

None.