

**WG14 N2179**  
**Meeting notes**

**C Floating Point Study Group Teleconference**

2017-09-26  
9 AM PDT / 12 PM EDT

**Attendees:** Rajan, Jim, Fred, Mike, David H., David C.

**New agenda items:**

None.

**Last meeting action items:**

Rajan: Check with David Keaton to see if SD3 will be considered in the next WG14 meeting and if so, can the floating point proposals (other than part 1 and 2) be postponed unless we are prepared for them. - Done.

Jim: List the proposals we have in flight with WG14 and what we need to do with them (if anything). - Done.

Jim: Create a new DR against part 3 for the non-arithmetic interchange formats to create format `_DECIMAL_DIG` type macros. - Done.

Jim: Augmented add `tgmath`: Change "invokes a real function" to "invokes a function returning a real type". - Done.

Jim: Augmented add: Create a proposal for the new augmented precision functions including `tgmath` for IEEE 2018 C binding. - Done.

Jim: Create a note to say the existing `fmin/fmax` functions may not correspond to the new IEEE 754:2018 `f{min/max}*` functions in TS Part 1 as part of the 2018 binding update. - Done.

**New action items:**

Jim: Activities in progress: Obsolesce `DECIMAL_DIG`: No proposal needed as our last one was accepted (N2108).

Jim: Activities in progress: `%a` for decimal differing from binary: Change "paper" to Reflector message 14792.

Jim: Activities in progress: Change all TS Part actions to "update and re-present" instead of just re-present.

Jim: Get document number and submit `DECIMAL_DIG` and `DIG` macros for non-arithmetic formats paper for the WG14 mailing.

Jim: Write a DR for the `_Roundwise` in `cbirt` sample for the WG14 mailing.

David H, Mike: Make a list of what has changed or been augmented in IEEE 754:2018.

Jim: Augmented arithmetic: F.10.13 is missing the paragraph number in the second paragraph.

Jim: Augmented arithmetic: F.10.13.3#4: computation `t` underflows -> computation of `t` underflows.

Jim: Min/Max functions: Mention that the existing fmin/fmax functions do not distinguish between +0 and -0.

Jim: Draft a response to Willem's WG14 paper (N2166) for next meeting.

**Next Meetings:**

Tuesday October 17th, 2017, 12:00 EDT, 9:00 PDT

Same teleconference number.

**Discussion:**

IEEE 754 revision:

Discussion on the payload get and set functions. Not yet resolved.

Not clear the best choice for rounding for the augmented precision functions. Most important for higher precision. If there are other use cases, it will help knowing them.

C++ liaison:

No update.

Activities ([http://wiki.edg.com/pub/CFP/WebHome/in\\_flight-20170916.pdf](http://wiki.edg.com/pub/CFP/WebHome/in_flight-20170916.pdf)):

\*Jim: Activities in progress: Obsolesce DECIMAL\_DIG: No proposal needed as our last one was accepted (N2108)

\*Jim: Activities in progress: %a for decimal differing from binary: Change "paper" to Reflector message 14792.

IEEE-754:2018 update: Can start on draft of a paper, but don't send it in until it passes ballot.

\*Jim: Activities in progress: Change all part actions to update and represent instead of just represent.

For next WG14 meeting:

Obsolesce DECIMAL\_DIG:

Discussed already.

DECIMAL\_DIG and DIG macros for non-arithmetic formats:

Waiting for the document #. Should be in the mailing.

\*Jim: Get document number and submit DECIMAL\_DIG and DIG macros for non-arithmetic formats paper for the WG14 mailing.

%a formatting consistency:

Discussed.

\_Roundwise in cbrt sample (Joseph Myers):

\*Jim: Write a DR for the \_Roundwise in cbrt sample for the WG14 mailing.

Binding for IEEE 754-2018:

WG14 paper about updating to IEEE 754-2018:

\*David H: Can make a list of what has changed or been augmented with Mike. Will need a WG14 person to translate it into WG14 wording.

Functions for augmented arithmetic

([http://wiki.edg.com/pub/CFP/WebHome/augop\\_spec-20170915.pdf](http://wiki.edg.com/pub/CFP/WebHome/augop_spec-20170915.pdf)):

Part 4 update.

\*Jim: Augmented arithmetic: F.10.13 is missing the paragraph number in the second paragraph.

\*Jim: Augmented arithmetic: F.10.13.3#4: computation t underflows -> computation of t underflows

Should we reconsider this being IEEE only?

Any other implementations have non-IEEE and are up to date with the standards?

Can do general wording with no specific rounding in the main body of the standard and then specify the rounding method in Annex F.

Will wait for 754 to decide what they will do with these functions.

Min/max functions ([http://wiki.edg.com/pub/CFP/WebHome/min-max\\_spec-20170924.pdf](http://wiki.edg.com/pub/CFP/WebHome/min-max_spec-20170924.pdf)):

\*Existing fmin/fmax don't distinguish between +0 and -0.

Payload functions:

If IEEE removes these functions, should they be removed from Part 1?

Mike: Decimal has the payload well defined. N-1 digits. How the payload is used is undefined.

Jim: Implementations can use some of the bits in the payload for themselves while some bits can be for the user. The definition of the payload is for diagnostic information in the glossary in the 754 spec.

Not clear if the user has access to everything in the payload due to the wording implying diagnostic information in there as well.

Don't want to conflict with any hardware or software payload values if the user wants to set their own.

Will wait to see what 754 does.

Other:

Willem Wakker evaluation formats (N2166):

Problem 1) Agree that evaluation format should be defined.

Not sure about where the text goes (language vs environment).

Seems to fit better where it is as part of the floating point model.

Problem 2) The change in DR500 seems to be fine and it is not unclear.

Characterization vs defined: Due to the use of characterization when discussing a number of things relating to this in the standard (like number of digits).

Problem 3) Can see the potential problem, but not sure if it is characterized correctly with the assumptions given. Ex. sloppy programmers.

Note, sloppy programmers are the vast majority of programmers.

Problem 4) Having stricter behaviour in Annex F allows more portability and hence it does deal with IEEE.

There was discussion in WG14 for this and it was decided to allow wide, but was OK with restricting it in Annex F.

Don't want to change Annex F.

Fred: Some implementations do have wider return values for some types and not for all.

Jim: Don't think we ever intended for FLT\_EVAL\_METHOD to affect function returns.

Fred will follow up with clarifying this return value issue.

Jim: Need more words to get implicit conversions included with what is in 5.2.4.2.2#9 after DR500.

TC's:

6) Looks good in principle, but should not require a cast (cast is a runtime operation) so you need to use a static constant or hex floating point.

\*Jim: Draft a response to Willem's WG14 paper (N2166) for next meeting.