

**WG14 N2163**  
**Meeting notes**

**C Floating Point Study Group Teleconference**

2017-07-11  
9 AM PDT / 12 PM EDT

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

**New agenda items:**  
None

**Last meeting action items:**

Jim: Reword [http://wiki.edg.com/pub/CFP/WebHome/augop\\_spec-20170422.pdf](http://wiki.edg.com/pub/CFP/WebHome/augop_spec-20170422.pdf) to have the additions done in Annex F only. - Done

Jim: Work [http://wiki.edg.com/pub/CFP/WebHome/min-max\\_names.pdf](http://wiki.edg.com/pub/CFP/WebHome/min-max_names.pdf) into a specification for the main body of the standard. Also add a footnote saying for implementations without NaN's the functions have the same results. - Done

All: Look into any other teleconferencing facility (Ex. Cisco Webex, Skype, etc.). - Not done

**New action items:**

Jim: Send the note (2017/06/28) drafted by Jim to WG14 for DR9 %a precision concerns.

Jim: Re DR501: Make a new proposal for Part 3 for new macros (format-specific DECIMAL\_DIG like macros)

Jim: Check implications for tgmth with regards to the augmented precision functions.

Jim: Augmented precision: Add in text to state that the functions force a particular rounding (ignoring static or dynamic rounding modes and implementation supported rounding modes).

Jim: min-max: Add in a statement about the preferred exponent.

Jim: min-max: Add a reference to the corresponding Annex F section for NaN treatment in the fmaximum/fminimum/fmaximum\_mag/fminimum\_mag functions.

Jim: min-max: {fmaximum/fminimum/fmaximum\_mag/fminimum\_mag}\_num functions: They determine the number -> they return the number

All: min-max: Consider what to do for fmin/fmax functions in the C standard.

Fred: Summarize what goes wrong for FLT\_EVAL\_METHOD for the things Fred has tested.

**Next Meetings:**

Tuesday August 29th, 2017, 12:00 EDT, 9:00 PST  
Same teleconference number.

**Discussion:**

IEEE 754 revision:

Another month or two until ballot likely.  
Expecting 2018 publish

C++ liaison:  
No update.

C2x proposals:  
Part 5d (alternate exception handling):  
Leave it as is unless there is a proposal, some direction or implementation experience.

DRs:  
DR9 (tgmth, arith conversions): Latest proposal from us is there. Looks good.  
DR11 (a-style formatting): Issue with the precision specification (binary vs decimal)  
Recommend it not be changed as per last meeting discussion.  
\*Send the note (2017/06/28) drafted by Jim to WG14.  
DR501 (C, DECIMAL\_DIG):  
Let DECIMAL\_DIG be obsolesced.  
\*Make a new proposal for Part 3 for new macros (format-specific DECIMAL\_DIG like macros)

Binding for IEEE 754-2018:  
Augmented precision ([http://wiki.edg.com/pub/CFP/WebHome/augop\\_spec-20170619.pdf](http://wiki.edg.com/pub/CFP/WebHome/augop_spec-20170619.pdf)):  
h, t names OK? Yes.  
Fred: Any mention for tgmth here?  
Jim: No, not going to try to add these to tgmth.  
David H: Usually these functions are used with the widest hardware precision present. Type generic wouldn't add much here.  
Fred: We should add words to say why these are not in tgmth.  
\*Check implications for tgmth with regards to the augmented precision functions  
Note that round to nearest, tie away from zero. This means it is independent of the current rounding mode (static and dynamic), and the implementation does not necessarily have to have support for that rounding mode, though the functions must have the results as if it is there.  
Can add these notes to F.10.13.  
min/max operations ([http://wiki.edg.com/pub/CFP/WebHome/min-max\\_spec-20170614.pdf](http://wiki.edg.com/pub/CFP/WebHome/min-max_spec-20170614.pdf)):  
In the rationale (paragraph 1), state it would be problematic rather than unusable.  
Even though these are optional, they are intended to be mandatory the next revision.  
Fred: Does the quantum matter for the zeros?  
Jim: Missing a statement about the preferred exponent here.  
Seems to be not specified in IEEE 754 so far. To be discussed in the IEEE 754 meeting.  
For the max/min functions it mentions differing from the max/min\_num functions due to NaNs, but doesn't say what it does for NaNs.  
\*Add a reference to the corresponding Annex F section for this in the

fmaximum/fminimum/fmaximum\_mag/fminimum\_mag functions

\*{fmaximum/fminimum/fmaximum\_mag/fminimum\_mag}\_num functions: They determine the number -> they return the number

Should fmin/fmax be obsolesced? Been out for a while. Can propose that as well.

The specification (7.12.12.2/3) could also be loosened for NaNs?

Other:

FLT\_EVAL\_METHOD (Willem Wakker):

Likely most compilers do the constant evaluation wrong and should have FLT\_EVAL\_METHOD -1, though it may be correct for operations.

The TS will allow you to request FLT\_EVAL\_METHOD 0.

\*Fred: Summarize what goes wrong for FLT\_EVAL\_METHOD for the things Fred has tested.

floating vs floating-point:

Inconsistent usage throughout the C standard. Can maybe say floating-point is for things that follow the floating point model.

Constant rounding modes and tgmth (Joseph Myers) - Email from Jim on 2017/07/07:

Deals with the example for this (cube root function).

Change proposed in the email seems good.

---