

**WG14 N2167**  
**Meeting notes**

**C Floating Point Study Group Teleconference**

2017-08-29  
9 AM PDT / 12 PM EDT

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

**New agenda items:**

Discussion about upcoming WG14 meeting.

**Last meeting action items:**

Jim: Send the note (2017/06/28) drafted by Jim to WG14 for DR9 %a precision concerns. - Done. Sent to WG14 on 2017/08/14.

Jim: Re DR501: Make a new proposal for Part 3 for new macros (format-specific DECIMAL\_DIG like macros). - Done. Sent on 2017/08/16.

Jim: Check implications for tgmath with regards to the augmented precision functions. - Done. Sent on 2017/08/18.

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). - Done. Sent on 2017/08/21.

Jim: min-max: Add in a statement about the preferred exponent. - Done. Sent on 2017/08/17.

Jim: min-max: Add a reference to the corresponding Annex F section for NaN treatment in the fmaximum/fminimum/fmaximum\_mag/fminimum\_mag functions. - Done. Sent on 2017/08/17.

Jim: min-max: {fmaximum/fminimum/fmaximum\_mag/fminimum\_mag}\_num functions: They determine the number -> they return the number. - Done (rejected change). Sent on 2017/08/17.

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

Fred: Summarize what goes wrong for FLT\_EVAL\_METHOD for the things Fred has tested. - Done. Sent on 2017/07/14.

**New 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.

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

Jim: Create a new DR against part 3 for the non-arithmetic interchange formats to create format\_DECIMAL\_DIG type macros.

Jim: Augmented add tgmath: Change "invokes a real function" to "invokes a function

returning a real type".

Jim: Augmented add: Create a proposal for the new augmented precision functions including tgmth for IEEE 2018 C binding.

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.

### **Next Meetings:**

Tuesday September 26th, 2017, 12:00 EDT, 9:00 PDT  
Same teleconference number.

### **Discussion:**

IEEE 754 revision:

Some open issues still:

Payload functions wanted by some people. Late proposal.

Another set of augmented functions that round to nearest wanted by some people. Not suitable for reproducible summation. Did not seem to be too much support. For performance on some platforms.

C++ liaison:

No update.

C2x proposals:

Nothing new to create.

\*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.

\*Jim: List the proposals we have in flight with WG14.

DRs:

DR501 ([http://wiki.edg.com/pub/CFP/WebHome/DR\\_for\\_macros\\_for\\_non-arith\\_formats-20170816.pdf?twiki\\_redirect\\_cache=f9e2368fc8345a340634aff44271c2ab](http://wiki.edg.com/pub/CFP/WebHome/DR_for_macros_for_non-arith_formats-20170816.pdf?twiki_redirect_cache=f9e2368fc8345a340634aff44271c2ab)):

Add to the existing set of DECIMAL\_DIG macros to apply to the new formats.

We thought DECIMAL\_DIG covered this, but it doesn't. With obsolescent of DECIMAL\_DIG, this is needed to cover that hole.

\*Create a new DR against part 3 for the non-arithmetic interchange formats to create format\_DECIMAL\_DIG type macros.

Binding for IEEE 754-2018:

Augmented add ([http://wiki.edg.com/pub/CFP/WebHome/augop\\_spec-20170817.pdf](http://wiki.edg.com/pub/CFP/WebHome/augop_spec-20170817.pdf)):

Change looks good.

Email on 2017/08/18: tgmth implications:

The second addition seems to be worded weirdly since the wording for the function returning a structure is talking about the return type instead of the type of the function

directly.

\*Change "invokes a real function" to "invokes a function returning a real type".  
min/max ([http://wiki.edg.com/pub/CFP/WebHome/min-max\\_spec-20170821.pdf](http://wiki.edg.com/pub/CFP/WebHome/min-max_spec-20170821.pdf)):

Change 1 (AI): Good.

Change 2 (missing text): Good.

Change 3 (forward references): Good.

Change 4 (preferred quantum exponents): Good.

Determine -> Return: C Standard uses "determine" and doesn't use "return".

Result: Keep it as "determine".

fmin/fmax that are in the existing C standard: May need to remove this from part 1 via a DR.

David H: If no one has implemented  $f_{\text{min/max}}_{\text{mag}}$ , maybe just get rid of them.

Can obsolesce fmin/fmax and say they are equivalent to  $f_{\text{min/max}}_{\text{num}}^*$ . Part 1 right now makes them not equivalent. Need to change part 1 to make it work.

Annex F would need to change to handle SNaN's with these.

The new functions would have to be non-conditional features to replace the existing.

Can alternatively put in a note to the relationship to the new functions and keep the existing ones.

David: Can also say the existing fmin/fmax don't necessarily correspond to any of these new functions and should only affect SNaNs. Existing functions generally work for most people since they don't have NaNs. No sense in changing them.

\*Jim: Write a note to say the existing fmin/fmax functions may not correspond to the new IEEE 754:2018  $f_{\text{min/max}}^*$  functions.

Nothing else expected to add before ratification or balloting.

Seems a small enough change to get another ~10 years conformance if we make these changes to bind to IEEE-754:2018.

Other:

FLT\_EVAL\_METHOD (Willem's email):

Basic issue is problem with widening floating point constants even if explicitly suffixed.

Should we bring attention to this possible pitfall? Or should we change wide evaluation?

Seems to have been implemented by at least some compilers as per Fred's 2017/07/14 email.

Fred's recent response seems to indicate this case is covered.

Jim: The new words in DR500 seem to not make this the case.

Perhaps add a new statement to cover this or make it more clear?

floating vs floating-point:

We have enough things for WG14 to look at without bringing this up unless someone wants to tackle it.

Constant rounding modes and tgmath (Email from Jim on 2017/08/29):

Agree with creating a DR against Part 1 to add in the `_Roundwise` aspect.