

WG14 N2188
Meeting notes

C Floating Point Study Group Teleconference

2017-10-17
9 AM PDT / 12 PM EDT

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

New agenda items:

Fred's emails

Adding attributes to C paper

IEEE 754-2018 changes review

Fred: Missing a number of functions that were added. Ex. fadd that returns a float from double args.

David H: It was already there in the 2008 standard.

Last meeting action items:

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

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

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

Jim: Get document number and submit DECIMAL_DIG and DIG macros for non-arithmetic formats paper for the WG14 mailing. - Done (N2171).

Jim: Write a DR for the _Roundwise in cbrt sample for the WG14 mailing. - Done (N2178).

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

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

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

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

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

New action items:

Jim: Activities: Add in WG14 document numbers to the items that have them.

Rajan: %a precision: If the footnote is wanted by WG14, correct the typo (differs is several ways -> differs in several ways)

Rajan: %a precision: If the footnote is wanted by WG14, change "For example" into "For instance"

Jim: Respond to WG14 reflector message 14812 with the need to have it predictable and fixed

Jim: Send WG14 N1361 to the CFP group

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)

Rajan: Willem's paper point 4 (return value): No consensus but are OK with either way and if no change, we should note that function return format is not being affected by FLT_EVAL_METHOD

Fred: Get the latest C standard paper to the group

All: Discuss via email Fred's email on 2017/09/21 about Annex F the TS. Potential DR for part 1.

Jim: Send out alternate proposal incorporating Willem's points this group agrees on.

Next Meeting:

Tuesday November 14th, 2017, 12:00 EST, 9:00 PST
Same teleconference number.

Discussion:

IEEE 754 revision:

Not much change since last meeting.

Still debating on adding payload functions.

Proposing deferring.

Still debating changing rounding mode for the augmented functions.

Proposing freezing current wording.

Aside: Don't say "table" something as it means different things (opposite) in different countries or is ambiguous.

C++ liaison:

No update.

Activities (http://wiki.edg.com/pub/CFP/WebHome/in_flight-20171004.pdf):

For updating to 754:2018: Just need a list of changes with some small description.

For next WG14 meeting:

Decimal Dig and Dig macros (<http://wiki.edg.com/pub/CFP/WebHome/n2171.pdf>):

Looks good.

%a consistency (WG14 reflector message 14792):

Typo: differs in several ways -> differs in several ways

For example -> For instance

_Roundwise in cbrt (<http://wiki.edg.com/pub/CFP/WebHome/n2178.pdf>):

Clark asked (WG14 reflector message 14810) why SHALL vs MIGHT.

Likely due to ease of implementation and efficiency (performance for example).

It is this way to keep it predictable and consistent and reproducible.

The user has explicitly added the pragma so they want something fixed.

Continue sending DRs to Blaine as web form format.

Binding for IEEE 754-2018:

Focus on TC for the next WG14 meeting so no need to bring this up in the meeting.
Better to have 754:2018 finalized before submitting to WG14 anyways.

Augmented arithmetic (http://wiki.edg.com/pub/CFP/WebHome/augop_spec-20170930.pdf):

Looks good.

After finalization of 754:2018, ensure it is still correct and in sync before presenting to WG14.

Fred: Should the augmul case call out what is invalid?

David: It should follow 754 since this a special case of the normal multiply (augmented).

Fred: Invalid is well defined in 754, but not in C.

Jim: We should only specify what 754 specifies for Annex F.

min/max (http://wiki.edg.com/pub/CFP/WebHome/min-max_spec-20170930.pdf):

Looks good.

Payload functions:

Leave as is.

Other:

Willem Wakker issues about evaluation formats

(http://wiki.edg.com/pub/CFP/WebHome/WW_evaluation_formats-20171007.pdf)

1) Jim: A lot of the floating point stuff is in section 5, not in 6.

2) The concern he had doesn't seem to apply to the actual wording after DR500.

Fred: If you have FLT_EVAL_METHOD as -1 then you don't know when it applies.

Jim: That's the point of -1.

Fred: The answer to Willem's question is "may" refers to the case where FLT_EVAL_METHOD is -1.

Jim: No, since FLT_EVAL_METHOD is the specific method of wide evaluation (if done).

3) Jim wrote a paper (N1361) to keep constants being affected by the evaluation format. It was already looked at by the committee and agreed to.

4) Fred: Already wrote a paper to have 'return' act like cast and assignment.

Jim: This was already discussed before and we decided to allow wider return values.

Ian: This would result in store and load on x86 (without SSE2 floating point - ABI incompatibility on register of return value).

The value has to be in the range of a float exponent and same precision for example, even if stored in a double register.

For our group, we don't have a position as we did bind to IEEE as per Annex F, and it is not our issue. It is more of a C issue.

There is a portability hole. It is unlike the other parts that are affected by FLT_EVAL_METHOD.

Jim: Without Annex F, if FLT_EVAL_METHOD is 0, then you know what happens with casts, assignments, parameters, etc. but not about function returns.

Fred: This was covered in C99 + TC3. It was taken out in C11. The argument was

it was added into C99 by accident.

Ian: It should be noted in the standard if nothing is done so it is clear to the standard reader that function return format not being affected by FLT_EVAL_METHOD is a hole.

For WG14, can say we don't have consensus but are OK with either way and if no change, to add the note as specified above.

Jim has an alternate proposal that incorporates what we agree with from Willem's paper. Will send it out to the group.

Other prep for WG14 meeting
C standard use of "floating" vs "floating-point"

All: Discuss via email Fred's email on 2017/09/21 about Annex F the TS. Potential DR for part 1.