WG14 N2331
Meeting notes

C Floating Point Study Group Teleconference

2018-11-28
8 AM PST / 11 PM EST / 4 PM UTC

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

New agenda items:
None.

Carry over 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)

Ian: There is an incompatibility. C++ has a macro to control the precision used where C leaves it to the implementation. Anything that shares a front end may have done the C++ way so C should copy and provide a control (as a should vs a shall). Without the macro they can do whatever they are doing today.

*Ian: Find the C++ standard reference and macro name for their handling of floating point literals.

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

David: Check the min/max C specification to ensure it matches what IEEE has. - Not done.

David: Check the augmented* C function specifications to ensure they match what IEEE has. - Not done.

All: totalorder* differ for NaN payloads: Note that we don't have approval to move up to 754 201x yet. - Keep open: Revisit after we move up to the 754 draft.

Last meeting action items:

Fred: Ensure that the items for P4_CR_for_rootn.pdf match IEEE. - Not done.

Jim: Create a CR for Part 4 from P4_CR_for_rootn.pdf. - Done.

All: Consider the printf for NaN(n-char-sequence) bounding issue.

New action items:

Rajan: See what FLT_EVAL_METHOD does with respect to constants in C++ in his implementations. - Overridden by the next item.

Ian: Find the C++ standard reference and macro name for their handling of floating point literals.

Jim: Let the WG14 editors know that we are waiting for the Part 1 integrated draft before putting in Part 2.

Fred: Provide words for a macro for printf n-char-sequence maximum length that the implementations have to define.

Jim: Provide the NaN payload specification editorial updates (positive signed floating point integers) to the WG14 editors.

Next Meeting(s):
Thursday, January 24th, 2019, 11:00 EST, 8:00 PST, 4PM UTC
Same teleconference number.
We now have sequence numbers for the emails!

Discussion:

754 revision:

Got the editorial coordination report back. Not too much of a problem other than wanting to sequentially number the tables and figures. Currently we kept the same numbering since 1985 for figures and 2008 for tables and references to them in emails so we want to keep it the same.

Need it signed off before starting the sponsored ballot.

C++ Liaison:

None.

C2X integration:

Part 1:

N2314 - TS 18661-1 + CLOSED CR/DRs + editorial changes + ISO template changes

N2315 - same as N2314, except with change marks relative to published TS 18661-1

Got positive response from the WG14 editors.

Part 2:

Jim: Done the same thing, but want to wait for the WG14 draft with the part 1 changes in it so can update the footnotes with the new draft.

*Jim: Let the WG14 editors know that we are waiting for the Part 1 integrated draft before putting in Part 2.

Fred: Do we still want the WANT macros since it is now in the C standard?

Jim: It is there right now since it is only library related.

Ian: For usability it is better to not require the user to specify a macro.

Rajan: Can create a macro to turn it off instead if wanted via a proposal.

Part 3:

Looks feasible. Work in progress.

Part 4ab:

Consider bringing this forward as an optional annex.

Part 5abcd:

Consider bringing this forward as an optional annex.

Action item details:

Ensure that the items for P4_CR_for_rootn.pdf match IEEE.

Fred to still verify.

Create a CR for Part 4 from P4_CR_for_rootn.pdf (http://wiki.edg.com/pub/CFP/WebHome/n2309.pdf)

Looks good.

Consider the printf for NaN(n-char-sequence) bounding issue (Fred’s 10/23 email “NAN(n-char-sequence)” and responses)

Jim: Mixed up two issues. Meant to talk about NaN() (which strtod accepts, but printf does not emit). Don't think this is a problem even though inconsistent. I don't think we need to make a change to this.

Fred: Just something I noticed. Not a real problem.

Other issues:

Update to suggested TC for CFP CR25 totalorder parameters (http://wiki.edg.com/pub/CFP/WebHome/Update_to_N2292_suggested_TC_for_CFP_CR25.pdf)

Updated to address the tgmath issue.

Looks good.

Fred’s 10/24 email “printf, NaN, infinity” and responses (Martin’s paper: http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2301.htm)

Fred: No way to specify which form the user wants. Also the lack of bound is another issue.

Jim: The n-char-sequence is implementation semantics. Don't want to force implementations to change. The security issue is valid. We can have a macro or change the specification to give a max limit for n-char-sequence for printf (not strtod).

*Fred: Provide words for a macro for printf n-char-sequence maximum length that the implementations have to define.

Rajan re WG14 meeting: 6.16 pow divide-by-zero case [N 2271]

Looks good to put in. It will be added to the minutes as an item that will go into C2X.

We will need to review this when it is done by the WG14 editors.

Rajan re WG14 meeting: 6.17 Min-max functions [N 2273]

FP study group to split this into two separate proposals. One for parts 1 and 2 changes based on the TS's, and the other for part 3 that fits the conditionally normative part 3 annex.

[Cfp-interest 1218]

http://wiki.edg.com/pub/CFP/WebHome/P1_CR_for_min-max_functions-20181127.pdf

http://wiki.edg.com/pub/CFP/WebHome/P2_CR_for_min-max_functions-20181127.pdf

Lets wait until the changes are in the C2X draft and look into making these proposals.

Fred's 2018/10/24 email "[Cfp-interest] FLT_ROUNDS and FE_TONEARESTFROMZERO"

Don't see this new rounding mode in part 1.

It was voted in by WG14 in N2124.

Jim: FLT_ROUNDS should be obsolesced since it only deals with additions.

Defer to next meeting.

Jim's 2018/11/01 email "payload spec for C2X integration draft" and Rajan's response

There were issues with saying unsigned integers. We fixed it later, but those changes were not sent to WG14.

*Jim: Provide the NaN payload specification editorial updates (positive signed floating point integers) to the WG14 editors.

Jim's 2018/11/09 email "footnote problems in TS part 1"

Looks good.