WG14 N2312
Meeting notes

C Floating Point Study Group Teleconference

2018-10-24
8 AM PDT / 11 PM EDT / 3 PM UTC

Attendees: Rajan, Jim, David H, Fred, 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)
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:
All: Consider the fact that C doesn't support the SNaN sequence that IEEE does. Can have strtod take it as input. - Done.
Rajan: Draft a paper on macro vs function (pointer vs arguments) causing signalling asking for recommendations from WG14 or ask if it is a problem for anyone. - Done.
Jim: Fix the sqrt and rootn identity conflicts with IEEE. - Done.
Fred: Ensure pown matches IEEE for the identity conflicts. - Done.

New action items:
Fred: Ensure that the items for P4_CR_for_rootn.pdf match IEEE.
Jim: Create a CR for Part 4 from P4_CR_for_rootn.pdf.
All: Consider the printf for NaN(n-char-sequence) bounding issue.

Next Meeting(s):
Wednesday, November 28th, 2018, 11:00 EST, 8:00 PST, 4PM UTC
Same teleconference number.

Discussion:
WG14 meeting: See Rajan's emails on October 16/17:
Should the part 1 and 2 updates include the editorial changes?
Yes.
Part 3 should be reworked into a conditionally normative annex to C2X.
For the SNAN parameter issue, it was a general statement.
Jim: Was the optional nature mentioned?
Rajan: Yes. Want macro and all.
Jim: Recommended is for specialized cases, if it was for general purpose use it would be required.

David: If something would make existing implementations non-conforming, it became "should" instead of "shall" to allow them to remain conforming.

Fred: Perhaps have a future directions section.

David: That would be a background document. There may well be one there.

Jim: There is also 'should' for some math functions for specialized uses. Perhaps have this as part of a background document?

Fred: There was a question asking if Jim has done any work on LaTeX for our docs?

Jim: No, I haven't.

Jim: For the new IEEE features, WG14 wants us to bring forward changes once IEEE is published and we can base the changes on C2X.

Fred: I have a paper on the precision selection for NaN payload free string output.

Rajan: CPLEX was cancelled as a project. Just as an FYI for the reduction operations.

754 revision:

They (IEEE standards association editors) have 30 days to comment. So far nothing.

Looking at things to change if we have a second ballot.

C++ Liaison:

Ian: Nothing new. Harder to talk to Hubert now. Specifically the issue is what C++ does for wide evaluation format in effect for constants (literals). Will likely get to it this afternoon.

Action item details:

Min/max C specification matches IEEE?

Nothing new.

Augmented* C function specifications match IEEE?

Nothing new.

totalorder* differ for NaN payloads: Note that we don’t have approval to move up to 754 201x yet.

Revisit after we move up to the 754 draft.

Consider the fact that C doesn't support the SNaN sequence that IEEE does. Can have strtod take it as input.

See Jim’s 10/9 email “[Cfp-interest] AI about C support for "snan" character sequences” and Mike’s responses

Jim: I thought 754 required reading of SNaN sequences, but on careful reading it was a recommendation, not a requirement. So we are conferment with 754.

754 has confirmed this interpretation in the last 754 meeting.

Paper on macro vs function (pointer vs arguments) causing signalling asking for recommendations from WG14 or ask if it is a problem for anyone.

Jim: Currently we have macros, functions that take pointers, etc. in the standard and the TS's. The main issue is on x87 is that it's hard to pass arguments without triggering Signalling NaNs. The interfaces have been there forever though.

Fred: SNaN support hasn't been required before this.

Jim: Do we need to do anything? Intel/legacy platforms can keep doing what they are doing and not be conforming.

Fred: WG14 can make it a 'should' instead of a 'shall' for fabs.

David: Most people using x87 probably expect it to stay as is. They know that likely no one will do it in the x87 way in the future. Leaving it as is will be a minor inconsistency for x87.

Fred: For existing functions with problems like fabs, make it recommended practice to not signal for SNaNs.

Jim: It's not a requirement in the main body that they not signal. Just annex F.

Fred: Make fabs a built-in operator is an option too.

Leave it as is for now.

Fix the sqrt and rootn identity conflicts with IEEE. http://wiki.edg.com/pub/CFP/WebHome/P4_CR_for_rootn.pdf

Jim: The cases listed are intended to be 1-1 to what's in 754.

@Fred: Ensure that the items for 4_CR_for_rootn.pdf match IEEE.

@Jim: Create a CR for Part 4 from 4_CR_for_rootn.pdf.

Ensure pown matches IEEE for the identity conflicts.

Jim: Fred (September 26th email) found one potential conflict (not a signaling nan case). 754 has it as a ballot comment to change it.

David: It's in the consensus comments.

Jim: With the change we are not in conflict.

C2X integration:

Nothing new.

Other issues:

Fred's 10/23 email "Three round nearest modes and FLT_ROUNDS" and responses

Jim: 754 intention is that roundsTiesToZero is not a general rounding mode.

Fred: It's not required as a general use. Should we get the spelling set?

Jim: No, since no one can since it is reserved.

Fred: The other issue is do we want a 1-1 mapping with FLT_ROUNDS as well?

Jim: FLT_ROUNDS already has the 4 modes for binary.

Fred: There is a fifth one for decimal.

Jim: I'd like to see FLT_ROUNDS go away since it only pertains to addition.

Fred: Annex F makes it apply to all operations.

Jim: Surely not.

Fred: F.2 says it applies to all types. Though not operations...

Fred to write a paper for this.

Fred's 10/23 email "NAN(n-char-sequence)" and responses

printf and strtod are different in that for printf if you have (you have to have n-char-sequence, where strtod has to accept no n-char-sequence.

Jim: Not sure there is a problem with that difference. strtod has to accept everything printf prints, but the strings can be from things other than printf as well.

Fred's 10/21 email "C DR/CR 432, 467"

See 10/23 FP model email from Fred for a better version.

Jim: For the second change, that second clause about fk digits seems to be trying to say too much in too little words.

Fred: It is eliminating the double-double case with the spaces between zeros.

Fred: Open to better words if anyone has them.

Jim: From the context, it seems you have already said what a floating point number is, and all that is left is what normalized is.

Fred: Not following.

Jim: Floating point types are able to represent "all of the" normalized floating point numbers. It is just writing the same as the previous statement. No need to say anything about the fk's there.

Fred: For paragraph 3, the k can be larger than the actual representation. p can be larger than what the hardware can actually do.

Jim: k can range between 1 and p.

Fred: The p in the model is not clearly the same as p in the hardware.

Jim: Saying positive or unsigned in parenthesis, it makes it seem saying zero is enough. Perhaps remove the parenthesis.

Fred: Could add a footnote saying the rest of the standard has positive, negative and unsigned zeros.

Jim: The last change for epsilon, is this the most useful definition?

Fred: There is also a math formula there.
Jim: Is it consistent with double-double?
Fred: As far as I know it is. Worked on the words along with the guy from IBM.

Fred's printf/NaN/Infinity email on 2019/10/24:
Prefer having the default be matching existing code.
@All: Consider the printf for NaN(n-char-sequence) bounding issue.
