WG14 N2304
Meeting notes

C Floating Point Study Group Teleconference

2018-09-25
8 AM PDT / 11 PM EDT / 3 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)
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:
Jim: Remove for quantum specification: "If x is NaN, the result is NaN". - Done.
Jim: Make the change to specify F.10.10a part 1 append as per Jim's binding meeting minutes email on 2018/08/23. - Done.
Fred: See which other functions have the need to not trigger signaling NaNs but are functions (need to be macros or have pointer parameters). - Done.
All: Look into comparison macros and how to work them to avoid SNaN's from signaling. - Done.
Jim: Create a part 1 CR to make the totalorder* functions take pointer arguments. - Done.
David: Look into the identity conflicts for sqrt and rootn in IEEE. - Done (in 754).
Jim: Part 4: Make the change as per Jim's 2018/06/26 email about specification for inexact with making the required operations raise inexact IFF it is inexact. - Done.
Jim: Part 1: Make the changes to next* as per Jim's email on 2018/08/20. - Done.
Fred: Give reference to the C DR for normalized double double meaning bits can be changed. - Done.

New action items:
All: Consider the fact that C doesn't support the SNaN sequence that IEEE does. Can have strtod take it as input.
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.
Jim: Fix the sqrt and rootn identity conflicts with IEEE.
Fred: Ensure pown matches IEEE for the identity conflicts.

Next Meeting(s):
Wednesday, October 24th, 2018, 11:00 EDT, 8:00 PDT, 3PM UTC
Same teleconference number.

Next WG14 meeting (Pittsburgh, 2018/10/15).

Discussion:

IEEE 754 revision:

Issues with getting a vote for the sponsored ballot.
Looks to be a 2019 revision.
There will be an update meeting after the ballot.
Requires the IEEE spreadsheet for comments.
Won't get an IEC # until after it is an IEEE standard.

C++ liaison:

Nothing.

Action item details (http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2288.pdf):

Min/max C specification matches IEEE?

Leave for next meeting.

Augmented* C function specifications match IEEE?

Not quite finalized yet.
Leave for next meeting.

Remove for quantum specification: "If x is NaN, the result is NaN". See Jim's 9/3 email "AIs to update working drafts" http://wiki.edg.com/pub/CFP/WebHome/cfp2x-20180903.pdf

Looks good.

Make the change to specify F.10.10a part 1 append as per Jim's binding meeting minutes email on 2018/08/23.

See Jim's 9/3 email "AIs to update working drafts" http://wiki.edg.com/pub/CFP/WebHome/cfp1x-20180903.pdf

Looks good.

See which other functions have the need to not trigger signaling NaNs but are functions (need to be macros or have pointer parameters).

See Fred's 8/28 email "Not trigger sNaN" and responses

Jim responded on the 9/16 email.

Mike: samequantum is intended to not trigger the SNaN. Intended to return a flag. Will discuss in the IEEE group.

printf: Fred: Does canonicalize trigger SNaN's since it takes a pointer?

Jim: Yes, it is a convert operation so it does (part 1 explicitly says it).

strtod/scanf: We don't say anything about this. Within C alone it is not a problem. Only when reading a character sequence from outside which may have an SNaN. Related to the NANS proposal.

Fred: IEEE-854 says the first characters for a nan should be NaN, but can be followed by whatever else. Note that this is incompatible with 754.

Look into comparison macros and how to work them to avoid SNaN's from signaling. Create a part 1 CR to make the totalorder* functions take pointer arguments.

http://wiki.edg.com/pub/CFP/WebHome/n2292.pdf

See also Joseph Myers 9/11 email "(SC22WG14.15517) totalorder and tgmath.h"

For totalorder, argument vs pointer, the intent seems to be for fast functions.

Fred: Seems to be a problem with Intel's float and double being promoted to long double which would cause a signal.

Fred: For fabs, it could be a BIF, but if the user code takes a function pointer to it, then any call through that function pointer would trigger the signal.

Fred: Can test this.

Jim: We could have an input SNaN macro if what we have specified is not practically implementable.

AI: 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.

Note that a CR should deal with the functions tgmath list fix-up

Look into the identity conflicts for sqrt and rootn in IEEE.

There was agreement in 754 to not change. That leaves it to us to change. We will need to expand out the cases.

AI: Jim: Fix the sqrt and rootn identity conflicts with IEEE.

Another identity conflict function was pown. We agree with IEEE there.

AI: Fred: Ensure pown matches IEEE for the identity conflicts.

Part 4: Make the change as per Jim's 2018/06/26 email about specification for inexact with making the required operations raise inexact IFF it is inexact.

See Jim's 9/3 email "AIs to update working drafts" http://wiki.edg.com/pub/CFP/WebHome/cfp4x-20180903.pdf

Page 16, ~line 10: Second sentence: Looks good.

Part 1: Make the changes to next* as per Jim's email on 2018/08/20.

See Jim's 9/3 email "AIs to update working drafts" http://wiki.edg.com/pub/CFP/WebHome/cfp1x-20180903.pdf

Page 42: Looks good.

Fred: Wanted to add the "if the result is finite, the value is normalized, sub-normal, or ..." to the main body of the standard.

Fred: It is not clear what nextup of 1 is on a double double system. Is it supposed to be representable in the model or the hardware?

Jim: Says the type of the function.

Ian: That would be the smallest positive sub-normal.

Give reference to the C DR for normalized double double meaning bits can be changed.

See Fred's 8/28 email "Normalized numbers"

Jim replied on 9/22, Fred replied to that on the same day.

CR 432: Looks good.

CR 467: p needs to be variable for double-double.

Normalized number: The rest of the sentence is still there.

Jim: The fk any digits part does not have to be there since it can't be any other way.

Fred: The double-double case is where you have the issue where the numbers are not part of the model, but allowed. Those numbers are not normalized. This change disallows them from being normalized.

Not intending to make double-double a part of the standard.

Fred: Not sure what would be most useful. It would at least be well defined.

Jim: I think it is well defined now. It would just be different from what is there now.

Fred: I think it is ambiguous now.

Other issues?

None.

Activities in progress:

Leave for later.