WG14 N2295
Meeting notes

C Floating Point Study Group Teleconference

2018-08-28
8 AM PDT / 11 PM EDT / 3 PM UTC

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

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.

Jim: Remove screen-share information from the agenda. - Done.

Fred: Recheck 'Functions and infinity' issues with 754 draft 240. - Done.

Last meeting action items:

David H: Check the new specification for inexact (action item 1) to make sure the new words still work with 754. - Not done. Will do it this meeting and close it.

Rajan: Bring a proposal forward for NaNQ and NaNS printf output to this group next meeting. - Done

Fred: Look into reduction functions and NaN's for optional exceptions. - In progress. On 754's plate now. Close it from our side.

New action items:

Jim: Remove for quantum specification: “If x is NaN, the result is NaN”

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

All: 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.

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

All: Look into comparison macros and how to work them to avoid SNaN's from signaling.

Jim: Create a part 1 CR to make the totalorder* functions take pointer arguments.

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

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.

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

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

Next Meeting(s):

Tuesday, September 25th 2018, 11:00 EDT, 8:00 PDT, 3PM UTC

Same teleconference number.

Discussion:

IEEE 754 revision:

Still in ballot in the microprocessor committee. Still a lot of discussion about going to sponsor ballot.

Mike is dealing with editorial issues.

At least 1 month before the ballot starts due to the editorial changes required. The ballot usually lasts a month.

Unlikely to be published this year. Should be ready by Spring WG14 meeting though.

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?

Leave for next meeting.

'Functions and infinity' issues with 754 draft 240. See Fred's 7/26 email "WG14 IEEE 754-C binding meeting minutes 2018/07/25" and responses.

See Jim's response on the 2018/08/23.

Unsigned infinity seems to be used when the sign does not matter. Our text is good. No conflict.

For quantum: why did we state NaN gives NaN since that is default.

AI: Jim: Remove for quantum specification: "If x is NaN, the result is NaN"

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

AI: All: totalorder* differ for NaN payloads: Note that we don't have approval to move up to 754 201x yet so what we have is only wrong when that standard gets out. We can revisit after we move up to the 754 draft.

totalorder* signature change to avoid signaling.

Fred: This applies to other functions as well such as samequantum.

Jim: Macros are OK since they are not really argument passing.

AI: Fred: See which other functions have the need to not trigger signaling nans but are functions (need to be macros).

Fred: Macro's don't help for expressions since you can't take the address of them. Ex. A function call that returns a SNaN as the return value as an argument to one of these operations.

Jim: Need to look at this more. I implemented these before and don't recall changing the compiler.

AI: All: Look into comparison macros and how to work them to avoid SNaN's from signaling.

Needs to be a part 1 CR.

AI: Jim: Create a part 1 CR to make the totalorder* functions take pointer arguments.

rootn: Jim: Formatting issue since 754 lists everything out but we use a summary statement.

Still that issue with rootn. There are a number of identities that would be good to be true, but some seem to be contradictory. We need to figure out which ones take priorities as a whole to avoid sqrt and rootn individually having clear solutions but contradictory together.

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

Regarding formatting for Annex F, we never tried to format it like 754. It is a big annex.

Rajan: Annex F is a C annex so should cater to that audience, not to us who need to compare the standards.

pown: CFP doesn’t has qNaN, why does 754 mention those?
Fred/Jim: It seems the general statement about NaN in = NaN out doesn’t work since it is specified otherwise here.
Leave it as is for now. If anyone feels like we need to change anything, bring it up.

reduction functions:
David: Can only short circuit if you have a SNaN.
David: Decided to leave sub-exceptions out since you couldn’t quit on invalid without getting all possible sub-exceptions for it.
Let 754 settle on what should happen and we can reflect that.

7.12.11 suggestion for samequantumdN: Any reason there is no SNaN specification here but is for others like this function?
Mike: Not sure. Been a while. Not useful if you have NaNs, but useful if you have finite numbers.
Fred: It is in the IEEE 754 2008 and the new one in draft status.
David: We said explicitly we don’t have any exception including signalling in 754.

Review of specification for inexact in Jim’s 6/26 email “AI about specification for inexact”.
Jim’s email on 2018/06/26: F.3 is only binding information. Decided to put it in F.10.1.7
Fred: Should we say for the first sentence that it doesn’t raise inexact otherwise? i.e. IFF
Agree to the change with changing the first part (required operations) to IFF.
AI: 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.

Proposal for NaNQ and NaNS printf output. See Fred’s 7/26 email “NaNQ, NaNS”.
Issues with parsing data if token based (would fail).
The last paragraph in 5.12.1 seems to allow this as an optional conversion.
Add in disallowing NANQ and possibly NANS for Annex F.

Reduction functions, optional exceptions.
Leave for later.

Issues with nextup and nextdown. See Fred’s 7/30 email “nextup()” and responses.
See Jim’s email response on 2018/08/20. Editorial changes.
AI: Jim: Part 1: Make the changes to next* as per Jim’s email on 2018/08/20.
For double-double: Ignore it since it is not IEEE.
AI: Fred to give reference to the C DR for normalized double double meaning bits can be changed.
Fred: Applies to nextafter as well.
Fred: Applications would probably expect normalized values for portability.

Issues with llogb. See Fred’s 8/3 email “llogb(1.)” and responses.
6.2.6.2#3 Doesn’t allow -0 other than the operations listed here so it should not be a problem.
Good for the llogb and ilogb cases.
Fred: OK with skipping it.
David: In 754 said we don’t talk about signed zero for integers at all.

Other issues?
None.

C2X Integration:
Mailing deadline in September.

Jim has access to the source repository for the standard.

Activities in progress:

Leave for later.