

WG 14 N1937
May 2015 CFP Teleconference Minutes

Attendees: Rajan, Jim, David, Mike, Ian, Vincent

New agenda items:

None.

Last meeting action items:

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 10: Note that we need to make sure functions with two or more arguments (since the order of evaluation of them is not fixed) is handled. - Listed as unspecified behaviour in IEEE. In our spec we say in Annex J the program has to deal with it. *Jim to send an email regarding this.

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 37: Look into tightening the underflow and inexact part. - Still open. - IEEE says the program can't depend on inexact or underflow. David: May be due to before or after rounding. *Keep open. David: With underflow, flags and traps are not the same due to exact underflow.

Jim: cfp5-20150330.pdf: p4: Line 20: FLT_EVAL_METHOD -> DEC_EVAL_METHOD - Done

Jim: cfp5-20150330.pdf: p4: Line 32: actually -> actual - Done

Jim: cfp5-20150330.pdf: p6: Line 23: Take into account rounding direction. - Already taken care of.

Jim: cfp5-20150330.pdf: p7: line 25 is misnumbered. It appears that one of the blank lines between 20 and 25 has a number. - Done

Jim: cfp5-20150330.pdf: p7: Line 10 is a line number on a blank line - Done

Jim: cfp5-20150330.pdf: p7: Last line: "NOTEIEC" appears to be missing a space. It has a tab. - Will leave it as is to see how it looks after publish edits.

Jim: cfp5-20150330.pdf: p7: Lines 31-32: Consider adding note about directed rounding - Seems to be taken care of already.

Jim: cfp5-20150330.pdf: p9: Lines 3 and 6: Consider clearer wording, maybe identifying IEEE/IEC section number - Done

Jim: cfp5-20150330.pdf: p10: Line 43: NOEXCEPT: Like default, but no flag is set: wording improvement needed. - Done

Jim: cfp5-20150330.pdf: p11: Line 3: OPTEXCEPT: Like default, but flag setting is optional: wording improvement needed. - Done

Jim: cfp5-20150330.pdf: p11: Line 4: Reference IEEE for the meaning of 'tiny'. Also look into 'preliminary' and exact underflow. - After the rewrite, not an issue.

Jim: cfp5-20150330.pdf: p11: Line 21: affect -> effect - Done

Jim: cfp5-20150330.pdf: p11: Line 29: Remove "undefined behaviour" statement for every action that has a "shall" requirement already (BREAK, GOTO, DELAYED_GOTO). - Done

New action items:

Ian: Talk to Michael Wong and Lowell regarding proposing this IEEE-754: 2008 binding to C++ as well.

Rajan, Jim: Talk to David Keaton regarding our intent for putting the essentials of parts 1 and 2 (not necessarily exact match) of this TS into the next C Standard.

Jim: Part 5: Write up the delayed goto transformation to prolog/epilog code.

Jim: Part 5: Write up text to allow 'break's to still cause the epilog code to be executed for delayed goto's.

Jim: Part 5: Add in examples of ASAP and Delayed Goto pragma's in the same block and in different orders.

Jim: Part 5: Add in a new issue - Is it worth adding expression evaluation methods that widen the library functions as well as the operators (that is already there)?

David: Part 5: Provide a mechanism (a new #pragma?) to allow implementations to possibly not propagate constant modes (rounding, exceptions).

Next Meeting:

June 11th, 2015, 12:00 EST, 9:00 PDT

Same teleconference number.

Note: This is Thursday!

Discussion:

Wiki cleanup: Older drafts need to be cleaned up on the wiki. Anyone can volunteer for this. Ex. Making a historical documents section. Meetings section as well. Topics page, etc. all need updating.

WG14 meeting:

Congratulations to Jim on the INCITS award for excellence for the work on these TS's.

IEEE binding parts 3 and 4 approved to go to publish after editorial review.

Part 5 looks good with some issues as will be discussed as per Jim's emails.

Should we propose to put the tech specs into the next C standard (or SD3)?

Parts 1-4 now and 5 later? 1-2 now, 3-5 later? All later? None (keep it as a tech spec)?

Jim: Better to put it into the C11 update (after LaTeX conversion) now since it is certain to uncover problems.

Should we have the proposal in early or later? By the Fall meeting or after that?

Given C usually does not take unimplemented specifications, we should do Parts 1 and 2 since part 1 is implemented by Intel and GCC, and Part 2 is implemented by IBM. The level of conformance to the implementation is still likely not exact, but should be close or very close.

Perhaps just state the intent we want some/all of this into the C standard.

This should be proposed for C++ as well. *ToDo: Ian to talk to Michael Wong and Lowell regarding proposing this binding to C++.

*ToDo: Rajan to work with Jim to talk to David Keaton regarding our intent for putting essentially parts 1 and 2 (not necessarily exact match) of this TS into the next C Standard.

Part 1: Published.

Part 2: Published. Updated to the correct formatting and style. Cover page had an issue but that is being fixed. Date macro fixed as well. Second edition should be published very soon.

Part 3: Same process as part 2. Part 3 is ready for the review committee.

Part 4: Same process as part 2. Part 4 should be done by end of week and then the review committee will be set up.

Part 5: Various emails (Fwd: [Cfp-interest] CFP Teleconference - Tuesday, May 19, 05/18/2015 01:22 PM).

Issue 1: The IEEE spec seems to say (a) - handle the exception using default exception handling. The implementation can be with a transform to a prolog/epilog using flags as stated in the IEEE spec. That can answer these questions. See David's note for the expected transformation in pseudo-code.

For multiple goto's, the second delayed goto's pragma results in an inner prolog/epilog outside the first delayed goto's pragma's prolog/epilog so it will be handled first.

We should have the transformation listed as the equivalent effect in the text.

*ToDo: Jim: Write up the transformation as the delayed goto mechanism.

Issue 2: Apply the transformation and so the goto/function would be taken directly. Perhaps say the effect happens if we actually fall out from the bottom of the associated block.

Break's should be a normal exit from the block?

*ToDo: Jim: Write up text to allow break's to still cause the epilog code to be executed for delayed goto's.

Issue 3: Any inner/outer #pragma would take precedence as to exception handling as per the transformation to prolog/epilog. Note if ASAP goto is first, then delayed goto, ASAP still occurs since it is before the end of the block. *ToDo: Jim: Create some examples of these cases.

Issue 4: Try Catch + pragma syntax:

David: Having try before catch is nice.

Should we even have try/catch?

If so, should the try/catch be in the same curly brace set?

Should the curly brace come before or after the pragma?

Don't put it in due to matching try/catch blocks and exception lists, and dealing with scope, etc. issues

We can list this as an idea considered and rejected when part #5 is presented.

Issue 5: Similar to constant rounding direction modes.

Consensus: Yes, keep similar to constant rounding modes.

Sequence point rules seem to not require explicit calling out of argument conversion before the call.

*New Issue: Is it worth adding expression evaluation methods that widen the library functions as well as the operators (that is already there)?

Before you could use the _t declarations (ex. float_t) and use type generics to get the same effect as widening.

Should we have a way of having constant rounding or other modes apply to user functions as well?

*ToDo: David: Provide a mechanism (a new #pragma?) to allow implementations to possibly not propagate constant modes (rounding, exceptions).

Regards,

Rajan Bhakta
z/OS XL C/C++ Compiler Technical Architect
ISO C Standards Representative for Canada
C Compiler Development