

WG 14 N1947
June 2015 CFP Teleconference Minutes

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

New agenda items:

Email discussions to be brought in.

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. - Keep open

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. David: With underflow, flags and traps are not the same due to exact underflow. - Keep open

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

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. Need to look at reflector message 13739. - Keep open

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

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

Jim: Part 5: Add in examples of ASAP and Delayed Goto pragma's in the same block and in different orders. - An example with two delayed Goto's and text re ASAP was added. - Keep open

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)? - Email sent to discuss this yesterday - Keep open

David: Part 5: Provide a mechanism (a new #pragma?) to allow implementations to possibly not propagate constant modes (rounding, exceptions). - Email sent out (May 20th, 2015) - Keep open

New action items:

Blaine: Send out a DR process document so we can use that for Parts 1-4.

Rajan: Create an outline of features in parts 1 and 2.

Jim: Update the wiki to put in links to the test suites (from Mike) and possibly compiler manuals that address these parts.

Fred: Find out the expiry date for an ISO document.

Jim: Page 1: Fill in dates for parts 3 and 4.

Jim: Page 3 and 4: Make the changes as proposed in the email on 2015/06/10 by Jim

All: Do the FENV_ALLOW_* pragmas apply to complex types?

Jim: Add a conformance macro for alternate exception handling.

Jim: Word the goto description better to disallow jumping to a label for an exception that did not happen.

Next Meeting:

July 7th, 2015, 12:00 EST, 9:00 PDT

Same teleconference number.

Discussion:

Part 1: Published.

Part 2: 2nd edition has been published.

Part 3: Final drafts have been sent to INCITS and should be sent to ISO soon. Estimating July 1st publication.

Part 4: Final drafts have been sent to INCITS and should be sent to ISO soon. Estimating July 1st publication.

*ToDo: Blaine: Send out a DR process document so we can use that for Parts 1-4.

Incorporating the TS into the C standard (see Joseph Myers email on 2015/05/21):

IBM doesn't have Part 2 complete (no constant rounding modes for example).

GCC is mostly library for part 1.

We need an inventory of the features in part 1 and 2 and have implementers check off what they have.

*ToDo: Rajan: Create an outline of features in parts 1 and 2.

Is there a comprehensive test suite for this?

Mike has some test suites available on his website.

Fred has coverage for most of it.

*ToDo: Jim: Update the wiki to put in links to the test suites (from Mike) and possibly compiler manuals that address these parts.

Status of IEEE 754-2008:

Upcoming for renewal.

Use Mike's errata sheet and have that be a revision.

Mike: The items don't justify the full process.

Ian: Add in relative error for an add for example in a portable way. Perhaps a correctly rounded dot product, complete arithmetic, exact dot product, etc. $0 * \text{Inf}$ if Inf came from an overflow, should give 0 instead of NaN.

David: Re: $\text{Inf} * 0$: Not used much (already implemented on sparc)

Still looking for someone to chair this group.

Expiring is unlikely. A number of people have volunteered to be a part of the committee.

*ToDo: Fred: What is the expiry date for an ISO document?

Part 5: Various emails + <http://wiki.edg.com/twiki/pub/CFP/WebHome/cfp5-20150608.pdf>

*ToDo: Jim: Page 1: Fill in dates for parts 3 and 4.

Re email on 2015/06/10 by Jim: action item on expression evaluation methods and math functions

The eval method can change value now (if it sees a pragma).

Consensus is to adopt the changes as proposed in the email.

*ToDo: Jim: Page 3 and 4: Make the changes as proposed in the email on 2015/06/10 by Jim

Page 7:

Should there be a shorthand for the list of operations?

Doing it would be a large editorial change and likely not easier to read by anyone.

Fred's email: Do the pragma's apply to complex types?

For complex types at IBM, they use FMA so if the pragma's apply this would change the behaviour.

For complex, multiple and divide there is no requirement they are commutative.
ToDo: All: Do the FENV_ALLOW_ pragmas apply to complex types?
We should make this a suggestion rather than require it.
Note that the real part vs the imaginary part could cause different exceptions. Ex. complex divide gives underflow and overflow both.
IEEE does not cover complex operations other than add and subtract.
Ex. Not required to produce most exceptions for complex operations.
David's email: 2015/05/20:
Keep the issue open.
Should alternate exception handling be moved to a new part, part 6?
The original reason for having them together was to have the same mechanism for both, but that has been settled.
Another option is to have another conformance macro for exception handling.
Leave this open?
Having a conformance macro allows more implementation choice and increases the odds of acceptance.
*ToDo: Jim: Add a conformance macro for alternate exception handling.
#pragma vs try/catch:
#pragma seemed the least disliked option
Some people felt there was strong sentiment in WG14 against keyword/syntax changes, but not all
Page 11: Optional_flag: Should this be FLAG(s)_UNSPECIFIED instead? OPTIONAL_FLAG seems to be confusing for a naive user.
This means that IEEE specified flags set for an operation need not be set.
Note that this could mean you could set a flag where not specified. This was not the intent.
Perhaps OPTIONAL_FLAG -> FLAG_OPTIONAL
Need to keep the relationship between NO_FLAG and OPTIONAL_FLAG
Suggestion: NO_FLAG -> NO_FLAGS, OPTIONAL_FLAG -> ELIDABLE_FLAGS (or perhaps without the S's)
Consensus is to keep as is for now.
Page 12: goto: The label/goto has to be linked to only the ones with exceptions that has happened (not to ones that have not happened).
*ToDo: Jim: Word the goto description better to disallow jumping to a label for an exception that did not happen.
The descriptions here relate to the goto's with the same associated block.
We need general clarity on goto's in general. This applies to the static rounding modes as well.
*ToDo: Blaine: Send an email to continue the goto discussion and issues.

Regards,

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