N1725

WG14 CFP meeting minutes for the meeting of 2013/06/13

Attendees: Jim, Fred, David, Mike, Marius, Rajan, Ian

Note taker: Rajan

Agenda:
New items requested:
Want to discuss endianess as well
Bool and unary +

Old action items:
AI: Jim to update Part 1 changes in Part 2 - Done
AI: Send email regarding generic/traditional - Done. Lot of discussion to go over.
AI: Mike to respond to Jim's reset email - Done

New action items:
Mike: Check what 754 says and provide a recommendation for INF, (S)NAN for quantum
Jim: Add the quantum function after renaming quantexp to iquantexp. Jim to try wording this suggestion
Jim: Redo the document (applies to part 2 and 3) with the unsigned char arrays for encodings, implementation defined bit/byte mappings, each array element will have 8 bits of the encoding regardless of the array element size.
All: Once the draft update to n1722 is sent, this is the review section assignments:
Intro: David
Clauses 1-5 + Bibliography: Fred
Clauses 6, 7: Marius
Clauses 8, 9: Ian, Mike
Clauses 10, 11: Ian
Clauses 12, 12.1, 12.2: Fred
Clauses 12.3: David
Clauses 12.4, 12.4.1, 12.4.2: Mike, Marius, Jim
Clauses 12.5-12.7: Fred
Clauses 12.8: Jim

Next meeting: June 11th, 2013, 12:00 EST, 9:00 PDT

Part 1 is up for ballot. August 16th expiry.
We (WG14) cannot discuss the document while it is out for ballot and this is a WG14 meeting.

Part 2:
Naming for float, double, long double:
WG14 was looking at "standard" as the term for them (see May 31st email).
Could also follow the 754 basic types name
That has a different meaning in C
Is "mandatory" good?
Anything other than "standard" would likely cause problems
Other ones are "conditionally mandatory/required"
"standard" may imply 754 types
We agree to keep "standard" and move forward
Should we call it "standard real floating types" or just call it "standard floating types"?

We should keep "standard floating types"

This and "decimal floating types" will collectively give the "real floating types" which fits into the C standard

IEC 60559 floating types naming scheme

Helps with part 3 as well. Should we do this?

Mike: Adding another layer of complexity without adding benefit unless we add binary types as well

We will go with the first method and not add the IEC 60559 to the type name classification

Quantum function:

We should have named the quantexp to iquantexp and used quantexp for the floating point version.

An alternative is to create a quantum function (using ilogb, you can get the quantexp function equivalent).

Suggestion from Mike: Add the quantum function after renaming quantexp to iquantexp.

AI Jim to try wording the suggestion from Mike above.

What should the quantum function do for the sign? Either always positive or the sign of the input?

Do what is in 754.

Quantum of INF? INF

AI Mike to check what 754 says and provide a recommendation for INF, (S)NAN

For part 2, we can split up the document in parts and have people focus on a particular part for review.

From n1722 (also in the wiki) *AI* once the draft is sent, this is the review section assignments:

Intro: David

Clauses 1-5 + Bibliography: Fred

Clauses 6, 7: Marius

Clauses 8, 9: Ian, Mike

Clauses 10, 11: Ian

Clauses 12, 12.1, 12.2: Fred

Clauses 12.3: David

Clauses 12.4, 12.4.1, 12.4.2: Mike, Marius, Jim

Clauses 12.5-12.7: Fred

Clauses 12.8: Jim

We need a draft for WG14 by late July, so we need to finish the review in the first week of July.

Part 3:

The non-arithmetic types issue:

Recent correspondence indicates unsigned char arrays for the encodings.

Group agrees on the general idea.

Note on June 10th:

Minor changes: const addition to function signitures and return type of int instead of void for the strfrom functions to match the existing strfrom functions for the other types we added

Endianness: 754 thought it was outside the scope to define it for floats and not ints

Ian: There are middle cases as well (some big some little between words and in a word for example)

Mike: We should avoid endianness in general

Niether 754 nor C deal with endianess, so we should not either

Mike: uint8_t should be used instead of unsigned char

Jim: Not a required type by an implementation

We need to make sure we have or acknowledge 8 bit bytes

Jim: We could require support for uint8_t

Fred: Posix requires 8 bit bytes

No sentiment to require 8 bit bytes

Fred: We should make the mapping between bits and bytes implementation defined

AI Jim to redo the document (applies to part 2 and 3) with the unsigned char arrays for encodings, implementation defined bit/byte mappings, each array element will have 8 bits of the encoding regardless of the array element size.

Note that these changes need to go into part 2 as well.