J11/WG14 Minutes (Revised)
N961
April 23-26, 2001
Copenhagen, Denmark

1 Opening activities

1.1 Opening Comments

Benito reminded the committee that under the timetable established two years ago that it was time for the Technical Report to enter the ballot process. Therefore, the committee should expedite producing a document for initial balloting.

1.2 Introduction of Participants/Roll Call

J11/United States:
John Hauser, Berkeley Design Tech
Tom Kremer, Cray
Frank Farance, Farance Inc.
Raymond Mak, IBM
David Keaton
David Schwab, Oracle
John Benito, Perennial (Convener)
Peter Seebach
Bill Seymour
Randy Meyers, Silverhill Systems
Douglas Walls, Sun (HOD)
Fred Tydeman, Tydeman Consulting

Canada:
Raymond Mak, IBM (HOD)
Walter Banks, ByteCraft

Denmark:
Allan Frederiksen, Nokia
Jan Kristoffersen, RAMTEX (HOD)
Peder Mousten, RAMTEX

Netherlands:
Randy Marques, Atos Origin (HOD)
Willem Wakker, ACE

Norway:
Keld Simonsen, RAP (HOD)

UK:
Francis Glassborow (HOD)

1.3 Selection of Meeting Chair

Benito was selected as Meeting Chair.

1.4 Procedures for this Meeting

Meyers was selected as secretary.

1.5 Approval of Previous Minutes (N928)

The minutes were missing the following action item:
Action: Benito will produce TC1 and submit it for balloting.

The minutes were approved with the above amendment.

1.6 Review of Action Items and Resolutions

1.6.1 The following action items are closed:

Action: Benito will produce Technical Corrigendum 1 and submit it for balloting.

Action: Benito will register the C locale.

Action: Mak will write up an alternative algorithm for the answering YES/NO question intended to be answered by N925. (The result is N926.)

1.6.2 The following action items are still open:

Action: Keaton will coordinate review of rationale. Helpers Gwyn, Larry Jones, Schwab, Farance, Peterson, Leca, Tydeman (floating point).

ACTION Benito: Talk to WG14 members who are involved in following the time issues within WG15, to find somebody willing to summarize differences from the WG14 specification and write them up for distribution.

ACTION Tydeman: To update N919 in light of committee discussion.

1.7 Approval of Agenda (N939)

Lunch is 1PM-2PM.

The issues from Japan send to the email reflector (N942) are to be processed during the defect report agenda items.

1.8 Distribution of New Documents

1.9 Information on Next Meeting

See document N941.

Benito reported that less expensive hotels for those seeking alternate lodging can be found using the web site referenced in N941.

1.10 Identification of National Bodies

Six countries: Canada, Denmark, the Netherlands, Norway, United Kingdom, United States.

1.11 Identification of J11 voting members

Meyers informed Cray that they would become voting members immediately after the meeting was adjourned.

Meyers apologized for missing the last meeting due to a back injury.

J11 Walls/Tydeman: Move to restore vote to Meyers. No objections.

Out of 15 voting members, 9 members were present, which is a quorum. (Later, Farance attended the meeting, bringing the number of members present to 10.)

2 Rationale Editors report (N940)

N940 is a list of edits made in producing N937, the Rationale. Keaton is no longer Rationale Editor. Benito is acting editor. Edits should go to him.

The Rationale is currently in Word, which has trouble editing such a large document. Benito reported that Larry Jones suggested moving it to troff, which is used for the Standard. Jones has tools to allow a troff document to be published in text, postscript, or PDF.

Action: Francis Glassborow will investigate whether the use of style sheets would improve the ability of Word to handle the Rationale.

Action: Benito and Keaton will run long long words by Japan.

3 TC1 Status Report (N932) (SC22 N3207) (Benito)

TC1 is undergoing SC22 ballot (ending May 20). All national bodies should have seen it. TC1 is expected to be approved.

Action: Benito will ask Larry about incorporating TC1 into the body of the Standard.

There were no objections to Benito asking SC22 for permission to publish TC1 freely.

Action: Benito will ask permission to publish TC1 freely.

4 Liaison Activities

4.1 J11

TC1 is in letter ballot, and will be discussed during the TAG meeting.

4.2 WG14 (SC22 N3206)

WG14 in Toronto asked Benito to forward C locale for registration with ISO/IEC 15897.

4.3 J16/WG21

Meeting next week in Copenhagen.

Working on TR on performance. First draft in mailing. Second draft coming soon, and will be discussed at meeting.

Working on TC1, expected to vote out at the next meeting.

Vote is expected on new work item for library extensions.

4.4 WG15 (POSIX)(N933, N934, N935) (Simonsen)

Action: Banks and Kristoffersen will review device I/O proposal.

POSIX Standard incorporating C99 and TC. Final Standard out for ballot June or July. One big standard for IEEE, Open Group, ISO.

4.5 WG20 (internationalization) (Simonsen)

Unicode 3.1 published in March contains some issues for C.

No objection to an agenda change to talk with representative of Unicode

4.6 Other Liaison Activities

4.6.1 Austin Group (formerly X/Open,) (Tydeman)

Tydeman has been working on updated math documentation for the Austin Group, and several issues arose which are being presented to J11/WG14.

4.6.2 WG11 (Wakker)

LID revision ISO 11404 is up for revision. Farance is project editor.
LIA, part 2 out for FDIS ballot.
LIA, part 3 work expected to start before SC22 plenary.

5 Fixed Point Arithmetic (N938) (Hauser)

SV Should overflow behavior (saturation, “mod”) be part of the type?
lots

A few people express vague concerns about wrap around arithmetic.

SV Who is in favor of prohibiting mixing fixed point and floating point?
7-3-7

SV Who is in favor of integer operands in + - * / not promoting to fixed point before the operation?
15-0-1

SV Who is in favor of requiring the basic arithmetic operations to calculate the result exactly when possible?
14-0-2

SV Who is in favor of special exception for multiply result of 1 or –1?
15-0-2

SV In favor of using type generic functions?
11-0-6

No objections to round
No objections bits like functionality

SV In favor of function like syntax on left hand side of assignment for `bits()`?
1-12-4

SV In favor of making it a constraint to arithmetically combine an accum and a frac when the fractional part of the accum is not big enough to hold all of the bits of fracc?
9-4-4

6 WDTR 18037 (N936) (Wakker, Kristoffersen, Banks)

N936 was presented.

7 Sequence Points N926 (Mak)

Meyers and Mak were in agreement that we should be careful to specify the semantics we want rather than the semantics that is simplest in the notation.

Meyers, Benito, Seymour found Mak's approach easier to understand than Feather's.

Glassborow reported that the UK desired a technical report with a formal, understandable model of sequence points, but did not have an official presence.

Action: Tydeman, Seymour, Seebach, Meyers will review/help.

ISO Web site has templates for the format of a technical report.

8 Defect Reports

8.1 DR 201

Accept the Proposed Committee Response and close DR.

8.2 DR 205

Accept the Proposed Committee Response and close DR.

8.3 DR 206

Accept the Proposed Committee Response and close DR.
Action: Benito make Rationale edit.

8.4 DR 207

Change the Proposed Technical Corrigendum: from: 6.4.1 append to paragraph 2:

"The keyword _Imaginary is reserved for specifying imaginary types such as described in Annex G."
to
"The keyword _Imaginary is reserved for specifying imaginary types.(footnote)
(footnote) One possible specification for imaginary types is Annex G."

Accept the Proposed Technical Corrigendum after edits and close DR.

8.5 DR 211

Accept the Proposed Technical Corrigendum and close DR.

8.6 DR 212

Action: Benito will reword the Committee Discussion to be more appropriate for the Record of Response, and will then make if available for review.

8.7 DR 214

Accept the Proposed Committee Response and close DR.

8.8 DR 215

Accept the Proposed Technical Corrigendum and close DR.

8.9 DR 217

Accept the Proposed Committee Response and close DR.

8.10 DR 218

Accept Proposed Committee Response and Proposed Technical Corrigendum and close DR.
Action: Benito make Rationale edit.

8.11 DR 219

Action: Meyers will write paper.

8.12 DR 221

Accept Proposed Committee Response and close DR.

8.13 DR 222

Spelling errors to correct: “shold” for “should” and “memeber” for “member”
Accept Proposed Technical Corrigendum. Do not include the Committee Discussion in the Record of Responses. Close DR.

8.14 DR 223

Action: Tydeman will revise Proposed Technical Corrigendum for later review.
Done.

8.15 DR 224

Action: Tydeman will revise Proposed Technical Corrigendum for later review.
Done.

8.16 DR 225

Accept Proposed Technical Corrigendum and close DR.

8.17 DR 226

Accept Proposed Committee Response and close DR.

8.18 DR 227

Accept Proposed Committee Response and close DR.

8.19 DR 229

Accept Proposed Technical Corrigendum and close DR.

8.20 DR 230

Action: Meyers will review.

8.21 DR 231

Accept Proposed Committee Response and close DR.
Action: Benito make Rationale edit.

8.22 DR 233

Spelling Correction: “otherwizw” for “otherwise”.
Change:
“Let P equal the precision of non-zero, 6 if the precision is omitted, of 1 if the precision is zero.”
to
“Let P equal the precision if non-zero, 6 if the precision is omitted, or 1 if the precision is zero.”
Accept Proposed Technical Corrigendum and close DR.

8.23 DR 235

Simonsen reported that on EBCDIC systems that strcoll did not behave like strcmp.

Action: Benito will contact venders to ask if their library behaves as proposed.
Done: Compaq reported that their implementation behaves differently.

Action: Mak will investigate whether IBM's libraries behave as proposed.
Done: IBM reported that their implementation behaves differently.

Close: No change will be made.

8.24 DR 236

Committee believes that Example 2 violates the aliasing rules in 6.5 paragraph 7:

"an aggregate or union type that includes one of the aforementioned types among its members (including, recursively, a member of a subaggregate or contained union)."

In order to not violate the rules, function f in example should be written as:

```
union tag {
  int mi;
  double md;
} u;

void f(int *qi, double *qd) {
  int i = *qi + 2;
  u.md = 3.1;   // union type must be used when changing effective type
  *qd *= i;
  return;
}
```

Example 1 is still open. The committee does not think that the suggested wording is acceptable.

8.25 Email Message 8335 (Issues from Japan)

Accept the answer to Item A in message 8336 from Jones as a proposed TC.

Action: Benito will inquire whether the answer to Item B in message 8336 from Jones is acceptable to the Japanese.

Action: Benito will inquire whether the answer to Item C in message 8340from Feather is acceptable to the Japanese.

Action: Benito will inquire whether the answer to Item D in message 8336 from Jones is acceptable to the Japanese. Note that the same notation is also used in paragraph 4 of Standard.

Action: Benito will inquire whether the answer to Item E in message 8336 from Jones is acceptable to the Japanese.

Action: Benito will inquire whether the following answer to Item F is acceptable to the Japanese:
"Normally, semantics rules only apply to the section that contain them. However, note that 6.5.16.2 paragraph 3 says that compound assignment is like simple assignment, and so this semantics rule applies to compound assignment as well. Rules for overlap for other contexts are given elsewhere in the Standard."

Action: Benito will inquire whether the answer to Item G in message 8336 from Jones is acceptable to the Japanese. Note that the corresponding binary operator is specified in 6.5.16.2 paragraph 3.

Action: Benito will inquire whether the answer to Item H in message 8336 from Jones is acceptable to the Japanese. See also N813 from the post-Colorado mailing.

Action: Benito will inquire whether the following answer to Item I is acceptable to the Japanese:
“Unless the implementation produces a result that is mathematically correct, the expression is not a constant expression. If overflow prevents the correct evaluation of the expression, the constraint is violated, and a diagnostic must be issued. The implementation is under no obligation to continue translating the program. A conforming compiler must issue a diagnostic or evaluate the expression correctly.”

Accept the answer to Item J proposed by the Japanese as a proposed TC.

Action: Benito will inquire whether the answer to Item K in message 8336 from Jones with the example from 8340from Feather is acceptable to the Japanese.

Accept the answer to Item L in message 8336 from Jones as a proposed TC.

Action: Benito will inquire whether the answer to Item M in message 8364 from MacDonald is acceptable to the Japanese.

Action: Benito will inquire whether the answer to Item N-1 in message 8336 from Jones with the example from 8340from Feather is acceptable to the Japanese.

Action: Benito will inquire whether the answer to Item N-2 in message 8340 from Feather is acceptable to the Japanese.

Proposed TC for Item O:

Change 6.10.3 paragraph 4:
If the identifier-list in the macro definition is not followed by an ellipsis, the number of arguments (including those arguments consisting of no preprocessing tokens) in an invocation of a function-like macro shall equal the number of parameters in the macro definition. Otherwise, there shall be more arguments in the invocation than there are parameters in the macro definition (note the ... is not a parameter). There shall exist a) preprocessing token that terminates the invocation.

Change 6.10.3 paragraph 12 to read:
If there is a ... following the identifier-list in the macro definition, then the trailing arguments, including any separating comma preprocessing tokens, are merged to form a single item: the variable arguments. The number of arguments so combined is such that, following merger, the number of arguments is one more than the number of parameters in the identifier list in the macro definition .

Action: Benito will inquire whether the answer to Item P in message 8336 from Jones is acceptable to the Japanese.

Action: Benito will inquire whether the answer to Item Q in message 8336 from Jones is acceptable to the Japanese.

Action: Benito will revise the following answer for Item R and present it to the Japanese:
“This is a syntax error since an undeclared identifier is not a primary-expression, it will cause a syntax error somewhere in the grammar because there is no place that the grammar allows either an identifier or a primary-expression.”

Action: Benito will inquire whether the answer to Item S in message 8336 from Jones is acceptable to the Japanese.

9 Fixed Point Issues

SV Who is in favor of prohibiting mixing fixed point and floating point (both constants or variables)?
4-12-3

SV Who is in favor of adopting a principle of range preserving rules for the usual arithmetic conversions (that it is better to preserve range at the expense of precision)?
lots-0-1

10 Email message 8375 (Tydeman)

Distributed SC22WG14 reflector as email message 8375.
The committee elected to make no change. This email message will not become a DR.

11 N943 (Tydeman)

Distributes as email message 8374.
Action: Benito will turn this document into DRs.
Action: Tydeman will contact IEEE arithmetic committee regarding remainder when dividing by zero.

Later in the meeting:

Walls provided feedback in message 8377.

Item 6 will become a DR to allow further discussion.

12 Benito Issues

Action: Benito will produce a DR about forming a composite type for array types using [static].

Meyers explained why bounds in variably modified types in casts must be evaluated. Consider expressions like: (int (*)[m]) ptr + 1. The value of m is needed so that the pointer can be incremented by the size of m ints.

13 TR Subgroup

Action: Wakker will produce the next version of the TR.

Banks presented background on multiple address spaces as segmented address spaces, and suggested that type qualifiers allowed elegant support in C.

SV Should work continue on named address space qualifiers for possible inclusion in the TR.
lots-0-1

14 Character Set Issues

Kent Karlsson, Specialist member, Unicode consortium and Member SC2/WG2, as representative from Sweden, discussed Unicode issues.

One issue was the Unicode consortium wanted to use wchar_t as a “multibyte-style” encoding for UTF16. (Microsoft has an API for this.)

Meyers and Mak said that was a violation of the programming model of wide characters, since a single wchar_t always represents a single character.

The committee also discussed that combining characters, and that uppercasing and lowercasing a string in some languages changes the number of characters in the string.

15 Specification-Based Datatypes (Farance)

Farance presented an overview of specification-based datatypes.

SV In favor of further investigation of specification based types?
12/1/4

16 Language Independent Datatypes (Farance)

Farance presented an overview of LID.

17 Reflector

SV Who is in favor of adding antispam software to the WG14 mail reflector?
12/2/6

18 Future Meetings

2001-10-15/19 Redmond, WA, USA (Microsoft)
2002-04-15/19 Curaçao, (AtosOrigin).

See the following for more information:
http://www.dkuug.dk/jtc1/sc22/wg14/www/meetings

Action: Schwab will contact Peterson to suggest a combined Oracle/Compaq sponsored meeting in Nashua, New Hampshire, USA during October 2002.

19 Mailings

The post-Copenhagen mailing deadline is May 25, 2001 at 5PM PDT.
The pre-Seattle mailing deadline is September 14, 2001 at 5PM PDT.
Documents for the mailing should be emailed to Benito. (jb@peren.com).

20 Adjournment

The meeting adjourned at 2:50pm, Thursday.

1 J11 TAG

1.1 Agenda

Appoint delegation and HOD for future WG14 meetings.
Discuss position on SC22 N3207 (TC 1)
Discuss ISO/IEC 14977:1996.
Discuss WG14 N944 (WG20 N822)

1.2 Delegation

J11 (Walls/Tydeman) Approve Jones, Walls, Tydeman, Seymour as delegation to the October 2001 meeting.
10/0/0/5/15

Meyers appointed Walls Head of Delegation.

1.3 Discuss position on SC22 N3207 (TC 1)

The issue is topic of a current letter ballot. Members who had not yet replied to the letter ballot communicated their votes to Walls. The letter ballot is expected to pass.

1.4 Discuss Syntactic metalanguage—Extended BNF (ISO/IEC 14977:1996)

J11 (Walls/Tydeman) Recommend "ISO/IEC 14977:1996 Information technology – Syntactic metalanguage -- Extended BNF" be approved for another 5 years.
2/2/6/5/15

1.5 Discuss Specification Method for Cultural Conventions (N944)

The committee chose not to take a position at this time.

1.6 Adjournment

The US TAG adjourned.

1 TR 18037 Subgroup Minutes

Meeting convened at Tuesday, 24 April 2001

Wakker elected chair.

Seymour appointed secretary.

1.1 Participants in the TR 18037 Subgroup

Bill Seymour	bsey@pobox.com
John Hauser	jhauser@bdti.com
Allan Frederiksen	allan.frederiksen@nokia.com
Willem Wakker	willemw@ace.nl
Jan Kristoffersen	jkristof@ramtex.dk
Walter Banks	walter@bytecraft.com
David Keaton	dmk@dmk.com

1.2 Agenda:

- Fixed point arithmetic
  - List of issues in Hauser's paper
  - Discussion of issues
    - Functionality
    - Syntax

- I/O
  - Separate memory spaces

- Other issues
  - Circular buffers

1.3 Issues (fixed-point):

1.3.1 The set of data types and constraints on implementations

Hauser: additional constraint for basic types: the number of integer bits in accum types either must be the same or must increase short->regular->long. Unsigned types have the same number of bits as signed types and range [0, 1), so the binary point is to the left of the left-most bit. The rationale is that one more bit of precision does make a difference when the size is only eight bits. Banks: Also important in fuzzy logic where degree of membership is always [0, 1).

Banks: Do we want <stdint.h>-like types? Consensus: yes, but accum types will need two sizes (number of integer & fractional bits).

Keaton: MIPS processors might have more fractional bits in the accumulator than in other registers. Hauser: it's easy for a compiler to ignore lower bits. Keaton: problem is in division. Hauser: DSP programmers usually expect division to be slow. Wakker: could we include some run-time mechanism to find out the number of fractional bits in accums? Hauser: it has always been the assumption that there would be macros for that similar to the <limits.h> macros.

Wakker: should conforming implementations support analogs of all standard types? Hauser: yes. Seymour: including long long? Consensus: no; but this could be an implementation extension.

Frederiksen: do we want fixed-point types to have the same sizes as integer types? Consensus: no.

1.3.2 C++ compatibility (of syntax)

Hauser: realizes that translation to C++ can be done easily...better to define the syntax in C-like fashion. [Secretary's note: this came up later in the full committee...it's likely to be contentious.]

1.3.3 Unsigned fixed-point types

Consensus: we should have them.

1.3.4 Is overflow control an attribute of the type?

Hauser: we don't want to invent some other type that the result will convert to, then demote to declared type. Nobody wants to attach e.g. saturation to the operator. Wakker: we could use something like a cast on an operator...we could have some pragma like FP_CONTRACT which could make a whole section of code do saturated arithmetic.

Options:

- attached to type

- attached to operator
  (sat)*
  *(sat)
  *e - new operator
  (*)
  <*>

- set mode with function call
  Hauser: problematic because some chips might not have the mode.

- set mode with pragma
  Hauser: pragma could control default case. Wakker: we should investigate what this means. Hauser: we know we want to select saturation at particular points...usually when demoting accum to fract.

- Frederiksen: have a sat function to set the mode. Hauser: if our implementation doesn't saturate for free, we might not want to do that.

- invent wider type, then have a sat operator
  Hauser: We need to allow sat on long accum, which could be the largest type.

Frederiksen: attached to type is acceptable work-around.

Consensus: allow both saturated types and pragmas; let the user choose. We don't want to require users to explicitly turn saturation on and off for each operation...that litters the code. Keaton: saturation really is a property of the type. [Secretary's note: Keaton made an analogy to pointers that I didn't record.] We'll need three pragmas: saturation on, modular wrap-around on, both off (default mode).

Hauser: can we have abbreviated casts like (sat)x instead of (fract_sat)x if we know x is fract? No way to turn it off, but that's less of a concern. Wakker: This is a language invention. Consider casting short to unsigned short...you can't just cast to unsigned. Consensus: don't do it.

Type names: sat/modwrap, saturated/wrapped. Real names will come from implementators' namespace. #include <stdfix.h> to get our names.

1.3.5 Expression evaluation

Same rules as for floating-point regarding associatively and order of evaluation? Consensus: yes.

Do we need support for modular wrap-around as well as saturation? Consensus: yes.

1.3.6 Mixing fixed-point and floating-point

Mixing fixed-point and floating-point...should it be a constraint violation?

Hauser: the reason we're writing fixed-point is that floating-point doesn't work in our CPU. Keaton: it would be surprising if the automatic conversion didn't work because one type can hold all the values of the other. Hauser: what's the harm? Keaton: it blows a hole in the model of the language, so we can't foresee the problems that will arise in the future. Trust the programmer. Hauser: no new arguments here...take back to committee. [Secretary's note: in full committee later, a straw poll opposed the constraint violation.]

1.3.7 Mixing signed and unsigned

Hauser: constraint violation. Binary point in different position, and unsigned has one extra bit of precision. Promotion to signed is acceptable because signed is "larger" (has larger range even though less precision). Keaton: doesn't agree that signed is "larger." Hauser: signed and unsigned seem to be separate markets. Banks: fuzzy logic will mix signed and unsigned; and probably would prefer to lose a bit of precision.

Hauser: a signed result is a reasonable answer, while an unsigned result is overflow if the signed operand is in fact negative. Keaton: promotion to signed violates the spirit of C, and would be more surprising to most programmers. Hauser: we don't want to be needlessly consistent. Keaton: comfortable with either promotion; but can't be a constraint violation. Banks: losing LSBs is better than losing sign, more like value-preserving because it preserves sign and MSBs, even unsigned long fract to signed fract. Consensus: bring back to committee. [Secretary's note: in full committee later, a straw poll supported the principle, where value-preserving promotions are not possible because neither type can hold all the values of the other, do range-preserving promotions.]

Do ints promote to fixed-point in expressions? Consensus: no.

1.3.8 Fixed-point arithmetic must be exact when possible

Wakker: not clear what it means. What do implementations have to do to get this right? Hauser: most existing implementations do nothing. Division is an interesting case, but probably won't be a huge overhead.

Farance: if we think in terms of an abstract machine that works with ints, then we get an easy way to describe the exactness requirement in standardese.

Wakker: in principle, we agree to require exactness, but wording will come later.

Keaton: there might be some future market where speed might be more important than accuracy.

Hauser: more concerned about what current processors do.

Consensus: let's accept this idea for now.

1.3.9 Make products of 1 or -1 exceptions to the exactness requirement?

Hauser: this exception is needed for a large number of known processors that do the wrong thing. Consensus: we don't like it; but we'll live with it for now.

Seymour: we can have recommended practice against it.

1.3.10 Rounding mode control

Hauser: if rounding is required, must always round up or down, but we might not know which. Most chips require extra work to do unbiased rounding (the 1/2-LSB case). Not as important for DSP applications. Do we want some standard way to specify rounding mode? Some chips will have a rounding mode so we'd want a function call to set the mode; others have no rounding mode so would need a pragma.

Farance/Keaton: if there's no prior art in DSP/fixed-point, use prior art in floating-point. Hauser: probably 60% to 70% of fixed-point processors have no mode bits. Farance: let's don't invent a whole new mechanism because we might get it wrong. Kristoffersen: let's investigate what the prior art really is. Hauser: most DSPs can't do what is done in floating-point, and it's more important to be fast than to gain another bit on average.

Banks: either ignore the problem or specify as in floating-point.

Consensus: table and research.

1.3.11 Type-generic functions?

Wakker: let's not open that can of worms. Hauser: too late...we have them now for floating-point.

Wednesday, 25 April 2001

1.3.12 bits operator

The objection in the full committee was to the lvalue version.

Wakker: could we have a conversion only to long accum, then cast to the proper type? What we need is two operators, one for conversion to int, and one for conversion from int. Consensus: yes.

1.4 IOHW discussion

Kristoffersen: changes to this version were made to address the comments from Ireland that expressed a preference for type-generic functions to many functions with suffixes on the names for many types.

This interface has been included in the C++ performance TR.

We need to take Annex C back to the full committee because of the angle brackets. These could be pragmas to get away from the look and feel of C++ templates; and such pragmas could be translated easily into C++ templates.

Kristoffersen gave an overview of his Annex C in N936. The idea is to provide enough standardization to allow source code portability, and to give some non-normative direction to implementors about how to define register access methods, and to suggest to vendors some ways that register access can be implemented.

Section C.5 of N936 will be broken out as a new annex D.

Hauser: what is the purpose of the functions in 3.2.3 for access_type initialization? Kristoffersen: this is mainly to provide a way for hosted systems to return a handle to the definition of an access_type. In embedded systems, the compiler would probably do this at compile time.

Hauser: what is the 3.1.3 access_type, e.g., MYPORT1 in the earlier example in 3.1? Kristoffersen: It's a statically defined definition of an I/O port contained in some header file. Seymour: is there a way for hosted implementations to define ports dynamically instead of statically? Kristoffersen: yes, see C.3, p. 28, "Based addressing" for an example.

Hauser: change "access type" to something else ("access spec macro" is suggested) because "type" already has another meaning in C.

Keaton: could functions like iord be function-like macros? Kristoffersen: yes, in fact the purpose is to encapsulate I/O register access in a portable way; and they must be function-like macros to allow compilers to generate inline code.

Hauser: can we take the address of an access spec macro? Banks: it's possible in my implementation.
Hauser: but we don't want that to be possible in general. Consensus: the access spec macro is a macro that can be used only as an argument to iord-like macros.

Changes to document:
- change "access type" to "access spec macro"
- change typedef to pragma
- in 3.2.3, make example more specific
- add more description in 3.1 about the three abstraction levels
- make C.5 a new annex D

1.5 Discussion of separate memory spaces (current C.5, future D):

Hauser: the term "device driver" has a very heavy meaning to some users. Let's use some other term.

Banks: we must deal with multiple address spaces, I/O address spaces, aliased memory, etc.

Hauser: how do you get pointers to these extended memory spaces?

Banks: the compiler must be smart enough to do the memory management. These really are variables in memory somewhere. What we're dealing with is C's inability to deal with multiple memory spaces.

Wakker: the problem is in the declaration of the variables. They should be declared as any other variable, then have a pragma that tells the compiler that the variable exists in some external memory space.

Hauser: it's confusing to have the ddram_r and ddram_w functions and all the associated comments in the document. This is part of the implementation that the user never sees.

Hauser: we don't want generic pointers because the access methods for different memory spaces might be different, or the processor might have a read instruction that requires the identity of the memory space.

Keaton: we could have fat pointers that contain information about access to other memory spaces. Banks: in a real-world case, this usually can be analyzed statically; and problems will arise only in pathological cases.

Hauser: user-defined type qualifiers aren't that complicated, but they will be a hard sell because they're a new language invention.

Keaton: this is a complicated issue that we can't resolve this week, so let's break this issue out into another paper or TR. Banks: this is an important topic that needs to go into the document as soon as possible.

Banks: "named address space" is a term for what we're talking about.

Seymour: is this like the prior art called "based addressing?"
Keaton: no, the "base" could be something like "56.101.202.5".
Banks: my Coke machine can access the memory in your Coke machine. It's really not an attempt to map extended memory spaces into a single linear space.

Hauser: suggests that we condone user-defined type qualifiers to identify external memory spaces, and put typemod in an annex.

Consensus: type qualifiers for extended memory could be implementation-defined, and in some implementations, user-defined.

1.6 Circular buffer discussion:

Frederiksen: DSP processors often allow specifying a memory area as a circular buffer to avoid memory accesses (vs. moving elements around in a linear array).

The functionality offered by various processors is so different that it's very difficult to come up with a consistent proposal.

Keaton: Could we have some kind of fat pointer that knows that it points to a circular buffer? Could such a pointer be declared to point only to a particular array?

Hauser: This can be done without any change to the language. The two options are some macro or library function that wraps a pointer:

```
for (...) {
   // ...
   p = CIRC( p + 1, /* array size info. */ );
}
```

Frederiksen: or use array indexing like:

```
for (...) {
   // ...
   a[ i % /* array size*/ ];
}
```

Compilers are already good at moving invariant calculations out of loops. They could recognize these idioms and load the DSP chip's circular buffer registers just as easily. Consensus: we don't recommend taking any action on circular buffers at this time.

1.7 Adjournment

Meeting adjourned at 15:10.