TS 18661 Part 5
Supplementary Attributes

WG 14 N1925

2015-04-10

IEC 60559 attributes

- N1919 – draft TS 18661 Part 5: Supplementary attributes
- First draft from FP study group
- Draft and presentation for early feedback

IEC 60559 attributes

- Constant modes for floating-point semantics
- Program specifies modes to apply to blocks
- Requires attributes for
  Rounding direction
- Recommends attributes for
  Evaluation formats
  Optimization control
  Reproducible code
  Alternate exception handling

C support for attributes

- Floating-point pragmas* in <fenv.h>
- Rounding direction pragmas in parts 1 and 2
- Pragmas for recommended attributes in part 5
- All similar in form and scope to STDC pragmas in C standard

* After email discussion about other syntax for alternate exception handling, believe unwise or unacceptable to introduce new syntax for FP

Evaluation formats

- #pragma STDC FENV_FLT_EVAL_METHOD *width*
  for standard and binary types
- *width* reflects a possible value of FLT_EVAL_METHOD macro (which characterizes default evaluation)
- Required support for *width* values -1, 0, and DEFAULT
- Other *width* values optional
- Similar FENV_DEC_EVAL_METHOD for decimal types
- Required support for decimal *width* values -1, 1, and DEFAULT

Optimization control

- Allow/disallow value-changing optimizations (transformations)
- #pragma STDC FENV_ALLOW_... *on-off-switch*
- VALUE_CHANGING_OPTIMIZATION allows all the following, which can also be allowed separately
- ASSOCIATIVE_LAW
- DISTRIBUTIVE_LAW
- MULTIPLY_BY_RECIPROCAL

  A / B = A x (1/B)

Optimization control (2)

- ZERO_SUBNORMAL

  allow replacing subnormal operands and results with 0
- CONTRACT_FMA

  contract (compute with just one rounding) A x B + C
- CONTRACT_OPERATION_CONVERSION

  e.g., F = D1 * D2 and F = sqrt(D)
- CONTRACT

  all contractions

  equivalent to FP_CONTRACT pragma in <math.h>

Reproducibility

- Support for code sequences whose result values and exception flags are reproducible on any conforming implementation
- #pragma FENV_REPRODUCIBLE *on-off-default*

FENV_ACCES	“on”
FENV_ALLOW_VALUE_CHANGING_OPTIMIZATION	“off”
FENV_FLT_EVAL_METHOD	0
FENV_DEC_EVAL_METHOD	1

Reproducibility (2)

Rules for reproducible code

- Translates into a sequence of IEC 60559 operations
- Under FENV_REPRODUCIBLE pragma
- Limits use of FP pragmas to reproducible states
- Not use long double, extended floating, complex, or imaginary types
- Use of part 3 interchange formats reproducible only among supporting implementations

Reproducibility (3)

Rules for reproducible code (cont.)

- Not use signaling NaNs
- Not depend on payload or sign bit of quiet NaNs
- Not depend on result value of conversion to integer type that would be “invalid” if the integer type had minimum allowed width
- Not depend on conversions between floating types and character sequences where character sequences are too long for *correct rounding*
- Etc.

Alternate exception handling

- IEC 60559 default exception handling
  - set exception flag(s)
  - return prescribed value
  - continue execution
- Way for a program to specify alternate exception handling

Alternate exception handling (2)

- #pragma STDC FENV_EXCEPT *except-list action*
- *except-list* a comma-separated list of

  exception macro names:

  FE_DIVBYZERO, FE_INVALID, FE_OVERFLOW, …

  and FE_ALL_EXCEPT

  and optional sub-exception designations:

FE_INVALID_ADD	inf - inf
FE_INVALID_MUL	inf * 0
FE_INVALID_SNAN	signaling NaN operand
FE_DIVBYZERO_LOG	log(0)
etc.	

Alternate exception handling (3)

action one of

- DEFAULT

  IEC 60559 default handling
- NOEXCEPT

  like default but no flags set
- OPTEXCEPT

  like default but flags may be set
- ABRUPT

  only for "underflow", IEC 60559-defined abrupt underflow shall occur, unlike ALLOW_ZERO_SUBNORMAL where zeroing may occur

Alternate exception handling (4)

action one of (cont.)

- BREAK

  terminate compound statement associated with pragma, ASAP*

- GOTO *label*

  jump to labeled statement, ASAP*

- DELAYED_GOTO *label*

  Complete compound statement associated with pragma, then jump to labeled statement

*ASAP – for performance, values and flags that might be set in the compound statement are indeterminate