SC22/WG14 N953

Additions/Comments on the first draft on
Extensions for the programming language C
to support embedded processors

Willem Wakker, ACE Consulting bv
September 2001

1 Fixed point constants

A *fixed-constant* is a *floating-constant* (see 6.4.4.2) without the *floating-suffix* followed by the *fixed-suffix*, defined as:

fixed-suffix: unsigned-suffix$_{opt}$ *long-suffix*$_{opt}$ *fixed-qual*

fixed-qual: one of
a A r R

The type of a fixed point constant depends on its *fixed-suffix* as follows (note that the suffix is case insensitive; the table below only give lowercase letters):

Suffix	Fixed point type
r	**fract**
ur	**unsigned fract**
lr	**long fract**
ulr	**unsigned long fract**
a	**accum**
ua	**unsigned accum**
la	**long accum**
ula	**unsigned long accum**

If the converted value does not fit in the internal representation of the indicated type (i.e., overflow occurs during the translation fase) the maximal or minimal value for the type, as defined in **<stdfix.h>** is stored.

Questions/discussion items:
1. A non-suffixed decimal constant can have a number of types, depending on its value: the constant has the type with the best range match (see table in 6.4.4.1). A similar approach could be followed (but is not proposed) for fixed point constants: if a constant with an **r** suffix cannot exactly be represented by a **fract** value, then the **long fract** type is a better match. However, unexpected (or unintentional) changes between sizes is considered to be a larger problem.
2. For the same reason as above, the even more elaborate type matching for hexadecimal integer constants is not copied.

2 <stdfix.h>

New constants are introduced to denote the behavior and limits of fixed point arithmetic.

A conforming implementation shall document all the limits specified in this section, as an addition to the limits required by the ISO C standard. The limits specified in this section shall be specified in the header file **<stdfix.h>.**

The values given below shall be replaced by constant expressions suitable for use in **#if** preprocessing directives.

The support for unsigned fixed point types is characterized by implementation-defined value of **unsigned_fixed_support**:

1 unsigned fixed point types supported
0 unsigned fixed point types not supported.

The values in the following sections shall be replaced by constant expressions with implementation-defined values with the same type. Except for the various **EPSILON** values, their implementation-defined values shall be greater of equal in magnitude (absolute value) to those shown, with the same sign. For the various **EPSILON** values, their implementation-defined values shall be less or equal in magnitude to those shown.

2.1 Sizes of fixed types

Note that for **unsigned_fixed_support** equals **0** (no unsigned fixed point support) the values for the unsigned types defined in this section are equal to the values of the corresponding signed types.

- number of bits for object of type **signed short fract**

  SFRACT_BIT 8

- minimum value for an object of type **signed short fract**

  SFRACT_MIN (-0.5r-0.5r)

- maximum value for an object of type **signed short fract**

```
SFRACT_MAX 0.9921875r                   // decimal constant
SFRACT_MAX 0X1.FCP-1r                   // hex constant
```

- the difference between **0.0r** and the least value greater than **0.0r** that is representable in the **signed short fract** type

```
SFRACT_EPSILON 0.0078125r               // decimal constant
```

```
SFRACT_EPSILON 0X1P-7r                  // hex constant
```

- maximum value for an object of type **unsigned short fract**

```
USFRACT_MAX 0.99609375ur                // decimal constant
USFRACT_MAX 0X1.FEP-1ur                 // hex constant
```

- the difference between 0.0r and the least value greater than 0.0r that is representable in the **unsigned short fract** type

```
USFRACT_EPSILON 0.00390625ur            // decimal constant
USFRACT_EPSILON 0X1P-8ur                // hex constant
```

- number of bits for object of type **fract**

```
FRACT_BIT 16
```

- minimum value for an object of type **fract**

```
FRACT_MIN (-0.5r-0.5r)
```

- maximum value for an object of type **fract**

```
FRACT_MAX 0.999969482421875r            // decimal constant
FRACT_MAX 0X1.FFFCP-1r                  // hex constant
```

- the difference between **0.0r** and the least value greater than **0.0r** that is representable in the **fract** type

```
FRACT_EPSILON 0.000030517578125r        // decimal constant
FRACT_EPSILON 0X1P-15r                  // hex constant
```

- maximum value for an object of type **unsigned fract**

```
UFRACT_MAX 0.9999847412109375ur         // decimal constant
UFRACT_MAX 0X1.FFFEP-1ur                // hex constant
```

- the difference between **0.0r** and the least value greater than **0.0r** that is representable in the **unsigned fract** type

```
UFRACT_EPSILON 0.0000152587890625ur     // decimal constant
UFRACT_EPSILON 0X1P-16ur                // hex constant
```

- number of bits for object of type **signed long fract**

```
LFRACT_BIT 32
```

- minimum value for an object of type **signed long fract**

```
LFRACT_MIN (-0.5R-0.5R)
```

- maximum value for an object of type **signed long fract**

```
LFRACT_MAX 0.99999999953433871269226074218751r
                                        // decimal constant
LFRACT_MAX 0X1.FFFFFFFCP-1lr            // hex constant
```

- the difference between **0.0lr** and the least value greater than **0.0lr** that is representable in the **signed long fract** type

```
LFRACT_EPSILON 0.00000000046566128730773925781251r
                                        // decimal constant
LFRACT_EPSILON 0X1P-31lr                // hex constant
```

- maximum value for an object of type **unsigned long fract**

```
ULFRACT_MAX 0.99999999976716935634613037109375ulr
                                        // decimal constant
ULFRACT_MAX 0X1.FFFFFFFEP-1ulr          // hex constant
```

- the difference between **0.0ulr** and the least value greater than **0.0ulr** that is representable in the **unsigned long fract** type

```
ULFRACT_EPSILON 0.00000000023283064365386962890625ulr
                                        // decimal constant
ULFRACT_EPSILON 0X1P-32ulr              // hex constant
```

2.2 Sizes of the accum type

Note that for **unsigned_fixed_support** equals **0** (no unsigned fixed point support) the values for the unsigned types defined in this section are equal to the values of the corresponding signed types.

- number of bits for object of type **signed short accum**

```
SACCUM_BIT 12
```

- minimum value for an object of type **signed short accum**

```
SACCUM_MIN (-8.0a-8.0a)
```

- maximum value for an object of type **signed short accum**

```
SACCUM_MAX 15.9921875a                  // decimal constant
SACCUM_MAX 0X1.FFCP+3a                  // hex constant
```

- the difference between **0.0a** and the least value greater than **0.0a** that is representable in the **signed short accum** type

```
SACCUM_EPSILON 0.0078125a            // decimal constant
SACCUM_EPSILON 0X1P-7a               // hex constant
```

- maximum value for an object of type **unsigned short accum**

```
USACCUM_MAX 15.99609375ua            // decimal constant
USACCUM_MAX 0X1.FFEP+3ua             // hex constant
```

- the difference between **0.0ua** and the least value greater than **0.0ua** that is representable in the **unsigned short accum** type

```
USACCUM_EPSILON 0.0078125ua          // decimal constant
USACCUM_EPSILON 0X1P-7ua             // hex constant
```

- number of bits for object of type **signed accum**

```
ACCUM_BIT 20
```

- minimum value for an object of type **signed accum**

```
ACCUM_MIN (-8.0a-8.0a)
```

- maximum value for an object of type **signed accum**

```
ACCUM_MAX 15.999969482421875a        // decimal constant
ACCUM_MAX 0X1.FFFFCP+3a              // hex constant
```

- the difference between **0.0a** and the least value greater than **0.0a** that is representable in the **signed accum** type

```
ACCUM_EPSILON 0.000030517578125a     // decimal constant
ACCUM_EPSILON 0X1P-15a               // hex constant
```

- maximum value for an object of type **unsigned accum**

```
UACCUM_MAX 15.9999847412109375ua     // decimal constant
UACCUM_MAX 0X1.FFFFEP+3ua            // hex constant
```

- the difference between **0.0ua** and the least value greater than **0.0ua** that is representable in the **unsigned accum** type

```
UACCUM_EPSILON 0.0000152587890625ua  // decimal constant
UACCUM_EPSILON 0X1P-16ua             // hex constant
```

- number of bits for object of type **`signed long accum`**

  `LACCUM_BIT 36`

- minimum value for an object of type **`signed long accum`**

  `LACCUM_MIN (-8.0la-8.0la)`

- maximum value for an object of type **`signed long accum`**

```
LACCUM_MAX 15.99999999953433871269226074218751a
                                        // decimal constant
LACCUM_MAX 0X1.FFFFFFFFCP+3la           // hex constant
```

- the difference between **`0.0la`** and the least value greater than **`0.0la`** that is representable in the **`signed long accum`** type

```
LACCUM_EPSILON 0.00000000046566128730773925781251a
                                        // decimal constant
LACCUM_EPSILON 0X1P-31la                // hex constant
```

- maximum value for an object of type **`unsigned long accum`**

```
ULACCUM_MAX 15.99999999976716935634613037109375ula
                                        // decimal constant
ULACCUM_MAX 0X1.FFFFFFFFEP+3ula         // hex constant
```

- the difference between **`0.0ula`** and the least value greater than **`0.0ula`** that is representable in the **`unsigned long accum`** type

```
ULACCUM_EPSILON 0.00000000023283064365386962890625ula
                                        // decimal constant
ULACCUM_EPSILON 0X1P-32ula              // hex constant
```