



**HAL**  
open science

## Contain the floating point naming explosion

Jens Gustedt

► **To cite this version:**

Jens Gustedt. Contain the floating point naming explosion. [Research Report] N2426, ISO JCT1/SC22/WG14. 2019. hal-02311460

**HAL Id: hal-02311460**

**<https://inria.hal.science/hal-02311460>**

Submitted on 10 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Contain the floating point naming explosion proposal for C2x

Jens Gustedt  
INRIA and ICube, Université de Strasbourg, France

The recent integration of the floating point TS has had a disastrous effect on the set of reserved identifiers in C. If C2x would go through as it currently stands about 1700 identifiers that were completely unprotected and unannounced would enter the global name space and nasty conflicts with user code would become unavoidable. There have been several attempts in history to introduce scoped identifiers to C, but we can't wait for such miracles to happen before C2x is supposed to come out. Therefore, we propose to take emergency measures and rename most of these new identifiers before they hit the public. On a complementary aspect, we propose decent names for the newly introduced floating point types.

### Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem analysis . . . . .	2
1.2	A simple way out (for WG14) . . . . .	2
<b>2</b>	<b>Strategy – Overview</b>	<b>3</b>
2.1	A first layer of changes . . . . .	4
2.2	A second layer of changes . . . . .	5
<b>3</b>	<b>Proposed changes</b>	<b>7</b>
3.1	Reserve the <code>_t</code> suffix for standard types . . . . .	7
3.2	Reserve the <code>_WIDTH</code> suffix . . . . .	7
3.3	Reserve the prefixes <code>DEC[0-9]+X?_</code> and <code>FLT[0-9]+X?_</code> . . . . .	8
3.4	Reserve the prefixes <code>dec[0-9]+x?_</code> , <code>flt[0-9]+x?_</code> , <code>flt_</code> and <code>ldbl_</code> . . . . .	8
3.5	Replace <code>d[0-9]+x?</code> suffixes by <code>dec[0-9]+x?_</code> prefixes . . . . .	8
3.6	Replace <code>f</code> and <code>l</code> suffixes by <code>flt_</code> and <code>ldbl_</code> prefixes . . . . .	9
3.7	Replace <code>f[0-9]+x?</code> suffixes by <code>flt[0-9]+x?_</code> prefixes . . . . .	9
3.8	Remove the reservation for the <code>cr</code> "prefix" . . . . .	9
3.9	Reserve more prefixes and suffixes . . . . .	9
3.9.1	Reserve the <code>fe_</code> prefix for <code>&lt;fenv.h&gt;</code> . . . . .	10
3.9.2	Reserve the <code>to</code> prefix for <code>&lt;math.h&gt;</code> . . . . .	10
3.9.3	Change the prefix for new complex functions to <code>cmplx_</code> . . . . .	10
3.9.4	Remove the <code>c</code> prefixed names from future library directions . . . . .	10
3.10	Change the <code>f</code> , <code>d</code> , <code>d[0-9]+x?</code> , and <code>f[0-9]+x?</code> prefixes for functions rounding to narrower type . . . . .	10
3.11	Rename some of the remaining features . . . . .	11
3.11.1	Rename the new floating point types . . . . .	11
3.11.2	Change the <code>FENV_</code> prefix to a <code>FE_</code> prefix. . . . .	12
3.11.3	Rename the new <code>&lt;fenv.h&gt;</code> features starting with <code>fe</code> to <code>fe_</code> . . . . .	13
3.11.4	Change <code>(u)?fromfp(x)?</code> identifiers to <code>to(u)?int(x)?</code> . . . . .	13
3.11.5	Change <code>llogb</code> to <code>toint_logb</code> . . . . .	14
3.11.6	Change <code>llquantexp</code> to <code>toint_quantexp</code> . . . . .	14
3.11.7	Change <code>SNAN</code> macros to <code>DBL_SNaN</code> and similar . . . . .	14
3.11.8	Change <code>CR_DECIMAL_DIG</code> to <code>FP_DECIMAL_DIG</code> . . . . .	15
<b>4</b>	<b>Current lists of new reserved identifiers.</b>	<b>16</b>

<b>5</b>	<b>Resulting lists of new systematically reserved identifiers.</b>	<b>24</b>
<b>6</b>	<b>Scripts for the transistion</b>	<b>31</b>
6.1	An sed replacement script . . . . .	31
6.2	An preprocessor header file for replacement . . . . .	60

**Note: List of identifiers in this paper will only be complete, once FP TS-3 has been integrated as Annex N.**

## 1. INTRODUCTION

As of this writing, a list of more than 1700 identifiers, see Section 4, following no established reservation rule, have been added to C2x. Among these identifiers are very short ones, such as **dadd**, **dfma**, **nanf32**, **pown**, or **SNAN**, that have a relatively high probability of conflict with user space identifiers. Others, such as **d64xadd128x** or **ufromfpxl** are very difficult to parse by the unprepared eye. Again others, such as the **totalorder** functions are “reserved by accident” because the **to** prefix is generally reserved for external identifiers.

Many of these identifiers are optional by implementation choices or must be explicitly requested by a “**WANT**” macro. Unfortunately, such mechanisms still reduce the design space for application libraries that aim to be portable. They still must avoid such identifiers, because they don’t know the settings in which their users will compile or link their library.

### *Making identifiers optional does not ensure portability.*

Such a long list of new names has some risk of conflict with existing code bases. Let us *e.g* assume that the probability  $p$  for conflict for each of these identifiers is one in a million for a particular source file. Assuming independence, the overall probability for all identifiers combined to have a conflict in a random source file would then be

$$1 - (1 - p)^{1700} = 1 - (1 - 10^{-6})^{1700} \approx 0.17\%$$

For large projects that are composed of many TU, this means that there is a relatively high probability of conflict. Such conflicts are not easy to detect because they may only manifest on platforms that are not accessible to the developer, maybe years after the code is written.

In contrast to that excessive intrusion into the user name space for the library, we then are shy to introduce properly usable names for the new features that really matter, namely the new floating point types. These are hidden behind new keywords and **typedef** (in Annex N) that use the impractical (and ugly) underscore-capital convention.

But, what is much worse that these discussions about probability of naming conflicts or other technical aspects, this careless addition of 1700 crude identifiers sends a very bad signal to our user base. Basically it says that we are willing to step on the feet of millions of C programmers and impact their everyday life for the next 30 years, because we, some dozen of implementers and language designers were not able to come up with a better naming scheme.

### *The C community deserves better than this.*

After the year-long acceptance problem of C99 by a large part of the C community, we have regained trust with C11 and C17. Let’s not jeopardize this lightly by throwing 1700 identifiers for mostly marginal features in the face of our users.

#### 1.1. Problem analysis

The addition of this abundance of identifiers mainly to the library is due to multiple factors, of which the principal are:

- The addition of the optional decimal floating point types and of the types of Annex N. For each existing (or added) functionality twelve additional new identifiers with suffixes

d128x	d32	d64	f128	f32x	f64x
d128	d64x	f128x	f16	f32	f64

are potentially added to `<math.h>`. Together with the already established suffixes `f` for **float** and `l` for **long double**, this means that for practically every `<math.h>` feature there may be fifteen functions (and linker symbols) and one macro in `<tgmath.h>`.

- For new functionalities, variants for **float** (suffix `f`) and **long double** (suffix `l`) are added systematically.
- In an attempt to reserve identifiers for future use, a non-functional set of identifiers starting with `cr` (for correct rounding) have been added to the future library directions. This convention is not even fit for internal use, because it is ambiguous, namely for the identifier `crsqrt` could be derived from `sqrt` (with a `cr` prefix) or from `rsqrt` (with a `c` prefix).
- Naming conventions are newly introduced (prefix `from`), instead of using established schemes to which users are accustomed and that are reserved to some extent by the standard (prefix `to`).
- Complicated naming conventions that depend of two types have been added, *e.g.* `d64muld128`, that use prefix and postfix notations.
- The addition of features with prefix `FENV_` or suffix `_WIDTH`.

Another aspect of all of this history of integration is that the mechanisms of C that are intended to guide users which identifiers might be used in future versions has failed badly for `<math.h>`. This is due to the fact that there are no established rules how identifiers would be added here.

*Each header should have a set of established rules for identifier addition.*

Generally, these rules should be expressed as easy-to-summarize prefix or suffix rules.

Finally, new keywords should only be introduced into the language for features that cannot be expressed syntactically, yet. In particular, new types are easily expressible in existing syntax, and so there should never be new keywords for types.

*New types should use reserved identifiers, not keywords.*

We already have an implicit rule for type names (suffix `_t`) we just have to establish it rigorously.

### 1.2. A simple way out (for WG14)

Seen all this inadequate handling of naming issues (introducing more than a thousand new linker symbols while weirdly reserving some dozens, completely marginal features, for future use) convinces me that the integration of at least part of the FP TS should be reconsidered. In its abundance of new names FP TS-3 is by far the worst, so I think the best and easiest for WG14 as a whole would be to send this back to the FP group.

We should not do this without giving the FP group directions about which (if any) other paths would be more promising and may have chances to be accepted by WG14.

The major problem of these interfaces is not the addition of names in some headers, it is the massive addition of symbols with external linkage that step on the feet of our users. Therefore, I think the primary goal should be a solution that seeks to minimize that effect. To my opinion, the best way forward would twofold:

- (1) Add new derived features (variant `fXXX` of function `f` for new type `XXX`) **only** through the type-generic macros.
- (2) Hide all additions of special types, in particular decimal floating point in specialized headers.

QUESTION 1. *Shall the integration of FP TS-3 into C2x be revoked?*

QUESTION 2. *Shall the integration of the FP TS be done that they are strongly backwards compatible, that is that they don't introduce new symbols in the user realm at any level (functions, macros, external linkage) unless the application source code requires it explicitly?*

QUESTION 3. *Shall all new additions of specialized interfaces for specialized floating point types be done via type-generic macros without introducing external symbols?*

QUESTION 4. *Shall all additions of new floating types be introduced in new headers?*

## 2. STRATEGY – OVERVIEW

If any of the above questions are answered affirmatively, the better, but unfortunately this cannot be all and we have to reflect about general strategies how to avoid such situations in the future and on how to repair C2x-current such that C2x-final will be acceptable by large parts of our user base. The following is based on a worst-case scenario where none of the strategies reflected in the questions above finds consensus. If otherwise, what is presented below would have to be adapted accordingly.

By a series of replacements and prefix or suffix reservations we try to get rid of most of the newly reserved identifiers, and replace them by more systematic choices that are easier to deal with. We introduce *a posteriori* rules that would have reserved these identifiers if they had been introduced in time.

*This paper only talks about identifiers that are introduced for C2x.*

If consequently there would be interest in renaming some C17 features, a different paper would have to be written.

In the following, we will use the following abbreviations for the corresponding types:

abbreviation	type
<code>flt</code>	<b>float</b>
<code>dbl</code>	<b>double</b>
<code>ldbl</code>	<b>long double</b>
<code>dec[0-9]+x?</code>	<code>_DecimalNx?</code>
<code>flt[0-9]+x?</code>	<code>_FloatNx?</code>
<code>int</code>	any signed integer type
<code>uint</code>	any unsigned integer type

### 2.1. A first layer of changes

If in a first pass we proceed as follows:

- Reserve the `_t` suffix for types (language and library).
- Reserve the `_WIDTH` suffix.
- Reserve the prefixes `DEC[0-9]+x?_` and `FLT[0-9]+x?_`.
- Reserve the prefixes `dec[0-9]+x?_`, `flt[0-9]+x?_`, `flt_`, and `ldbl_`.
- Transform all `f`, `l`, `d[0-9]+x?`, `f[0-9]+x?` suffixes into the appropriate prefixes.
- Remove the new `cr` identifiers.

We remain with a list of new main features that would be introduced by C2x:

acosf16	d128sub	f32mul	fmaxf16	<b>pow</b>
acoshf16	<b>d32add</b>	f32sqrt	<b>fmaxmag</b>	pownf16
<b>acospi</b>	<b>d32div</b>	f32sub	fmaxmagf16	<b>powr</b>
acospif16	<b>d32fma</b>	f32xadd	fminf16	powrf16
asinf16	<b>d32mul</b>	f32xdiv	<b>fminmag</b>	<b>quantize</b>
asinhf16	<b>d32sqrt</b>	f32xfma	fminmagf16	quantizef16
<b>asinpi</b>	<b>d32sub</b>	f32xm1	fmodf16	<b>quantum</b>
asinpif16	<b>d64add</b>	f32xsqrt	<b>fmul</b>	quantumf16
<b>atan2pi</b>	<b>d64div</b>	f32xsub	frexpf16	remainderf16
atanf16	<b>d64fma</b>	f64add	<b>fromfp</b>	rintf16
atanhf16	<b>d64mul</b>	f64div	fromfpf16	<b>rootn</b>
<b>atanpi</b>	<b>d64sqrt</b>	f64fma	<b>fromfpx</b>	rootnf16
atanpif16	<b>d64sub</b>	f64mul	fromfpxf16	<b>roundeven</b>
<b>cacospi</b>	d64xadd	f64sqrt	<b>fsqrt</b>	roundevenf16
<b>canonicalize</b>	d64xdiv	f64sub	<b>fsub</b>	roundf16
canonicalizef16	d64xfma	f64xadd	<b>getpayload</b>	<b>rsqrt</b>
<b>casinpi</b>	d64xm1	f64xdiv	getpayloadf16	rsqrtf16
<b>catanpi</b>	d64xsqrt	f64xfma	HUGE_VAL_F16	<b>samequantum</b>
cbrtf16	d64xsub	f64xm1	hypotf16	samequantumf16
<b>ccompoundn</b>	<b>dadd</b>	f64xsqrt	ilogbf16	scalblnf16
<b>ccospi</b>	<b>ddiv</b>	f64xsub	ldexpf16	scalbnf16
ceilf16	<b>deprecated</b>	fabsf16	lgammaf16	<b>setpayload</b>
<b>cexp10</b>	<b>dfma</b>	<b>fadd</b>	<b>llogb</b>	setpayloadf16
<b>cexp10m1</b>	<b>dmul</b>	fdimf16	llogbf16	<b>setpayloadsig</b>
<b>cexp2m1</b>	<b>dsqrt</b>	<b>fdiv</b>	<b>llquantexp</b>	setpayloadsigf16
<b>clog10p1</b>	<b>dsub</b>	<b>fe_dec_getround</b>	llquantexpf16	sinf16
<b>clog2p1</b>	erfcf16	<b>fe_dec_setround</b>	llrintf16	sinhf16
<b>clogp1</b>	erff16	<b>fegetmode</b>	llroundf16	<b>sinpi</b>
<b>compoundn</b>	<b>exp10</b>	<b>FENV_DEC_ROUND</b>	<b>log10p1</b>	sinpif16
compoundnf16	<b>exp10m1</b>	<b>FENV_ROUND</b>	<b>log2p1</b>	<b>SNAN</b>
copysignf16	<b>exp2m1</b>	<b>fesetexcept</b>	logbf16	SNANf16
cosf16	expf16	<b>fesetmode</b>	logf16	sqrtf16
coshf16	f128add	<b>fetetestexceptflag</b>	<b>logp1</b>	tanf16
<b>cospi</b>	f128div	<b>ffma</b>	lrintf16	tanhf16
cospif16	f128fma	floorf16	lroundf16	<b>tanpi</b>
<b>cpow</b>	f128mul	FLT16_DECIMAL_DIG	<b>maybe_unused</b>	tanpif16
<b>cpowr</b>	f128sqrt	FLT16_DIG	modff16	tgammaf16
<b>CR_DECIMAL_DIG</b>	f128sub	FLT16_EPSILON	nanf16	truncf16
<b>crsqrt</b>	f16add	FLT16_MANT_DIG	nearbyintf16	<b>ufromfp</b>
<b>crsqrt</b>	f16div	FLT16_MAX	nextafterf16	ufromfpf16
<b>csinpi</b>	f16fma	FLT16_MAX_10_EXP	<b>nextdown</b>	<b>ufromfpx</b>
<b>ctanpi</b>	f16mul	FLT16_MAX_EXP	nextdownf16	ufromfpxf16
d128add	f16sqrt	FLT16_MIN	nexttowardf16	
d128div	f16sub	FLT16_MIN_10_EXP	<b>nextup</b>	
d128fma	f32add	FLT16_MIN_EXP	nextupf16	
d128mul	f32div	FLT16_TRUE_MIN	<b>nodiscard</b>	
d128sqrt	f32fma	fmaf16	powf16	

This looks already a bit easier to handle. But still this has a lot of identifiers that are not even active and a lot that would profit from a more systematic approach.

## 2.2. A second layer of changes

A second layer is more targeted to correct specific naming problems:

- Introduce some more systematic reservations:
  - Reserve the `fe_` prefix for `<fenv.h>`.
  - Reserve the `to` prefix for `<math.h>`.
  - Change the `c` “prefix” for future complex functions to `cmplx_`.
- For the operations that convert to a narrower type, change the `f`, `d`, `d[0-9]+x?`, and `f[0-9]+x?` prefixes to already reserved prefixes `toflt_`, `todbl_`, `todec[0-9]+x?_`, and `toflt[0-9]+x?_`, respectively.
- Rename some of the remaining features:
  - Change the names of the decimal floating types to `dec32_t`, `dec64_t` and `dec128_t`.
  - Change the names of the interchange floating types (Annex N) to `flt32_t`, `flt64_t` and `flt128_t`.
  - Change the names of the extended floating types (Annex N) to `flt32x_t`, `flt64x_t`, `flt128x_t`, `dec64x_t`, and `dec128x_t`.
  - Change the names of the evaluation floating types (Annex N) to `flt32_eval_t`, `flt64_eval_t`, `flt128_eval_t`, `dec32_eval_t`, `dec64_eval_t`, and `dec128_eval_t`.
  - Change the `FENV_` prefix to a `FE_` prefix.
  - Rename the new `<fenv.h>` features starting with `fe` to `fe_`.
  - Change `(u)?fromfp(x)?` identifiers to `to(u)?int(x)?`
  - Change `llogb` to `toint_logb`.
  - Change `llquantexp` to `toint_quantexp`.
  - Change `SNANXXX` to `PREFIX_SNAN`.
  - Change `CR_DECIMAL_DIG` to `FP_DECIMAL_DIG`.
  - Change `HUGE_VAL_XXX` to `XXX_HUGE_VAL`.

The remaining list of non-systematic identifiers for new main features is then:

<code>acospi</code>	<code>CMPLXF64</code>	<code>exp10</code>	<code>maybe_unused</code>	<code>roundeven</code>
<code>asinpi</code>	<code>CMPLXF64X</code>	<code>exp10m1</code>	<code>nextdown</code>	<code>rsqrt</code>
<code>atan2pi</code>	<code>compoundn</code>	<code>exp2m1</code>	<code>nextup</code>	<code>samequantum</code>
<code>atanpi</code>	<code>cospi</code>	<code>fmaxmag</code>	<code>nodiscard</code>	<code>setpayload</code>
<code>canonicalize</code>	<code>decodebin</code>	<code>fminmag</code>	<code>pown</code>	<code>setpayloadsig</code>
<code>CMPLXF128</code>	<code>decodedec</code>	<code>getpayload</code>	<code>powr</code>	<code>sinpi</code>
<code>CMPLXF128X</code>	<code>deprecated</code>	<code>log10p1</code>	<code>quantize</code>	<code>tanpi</code>
<code>CMPLXF32</code>	<code>encodebin</code>	<code>log2p1</code>	<code>quantum</code>	
<code>CMPLXF32X</code>	<code>encodedec</code>	<code>logp1</code>	<code>rootn</code>	

These names of (new) features are those that users should memorize. By using type generic macros for the remaining floating point features, where possible, they might entirely avoid any complicated composed identifiers.

We obtain the following list of systematically reserved identifiers for the other new features:

DBL_SNAN	todbl_add	todec64x_fma	toflt32x_sqrt
dec128_t	todbl_div	todec64x_mul	toflt32x_sub
dec128x_t	todbl_fma	todec64x_sqrt	toflt64_add
dec32_t	todbl_mul	todec64x_sub	toflt64_div
dec32x_t	todbl_sqrt	toflt128_add	toflt64_fma
dec64_t	todbl_sub	toflt128_div	toflt64_mul
dec64x_t	todec128_add	toflt128_fma	toflt64_sqrt
decimal128_t	todec128_div	toflt128_mul	toflt64_sub
decimal32_t	todec128_fma	toflt128_sqrt	toflt64x_add
decimal64_t	todec128_mul	toflt128_sub	toflt64x_div
FE_DEC_ROUND	todec128_sqrt	toflt16_add	toflt64x_fma
fe_getmode	todec128_sub	toflt16_div	toflt64x_mul
FE_ROUND	todec32_add	toflt16_fma	toflt64x_sqrt
fe_setexcept	todec32_div	toflt16_mul	toflt64x_sub
fe_setmode	todec32_fma	toflt16_sqrt	toflt_add
fe_testexceptflag	todec32_mul	toflt16_sub	toflt_div
float128_t	todec32_sqrt	toflt32_add	toflt_fma
float32_t	todec32_sub	toflt32_div	toflt_mul
float64_t	todec64_add	toflt32_fma	toflt_sqrt
flt128_t	todec64_div	toflt32_mul	toflt_sub
flt128x_t	todec64_fma	toflt32_sqrt	toint
flt32_t	todec64_mul	toflt32_sub	toint_logb
flt32x_t	todec64_sqrt	toflt32x_add	toint_quantexp
flt64_t	todec64_sub	toflt32x_div	tointx
flt64x_t	todec64x_add	toflt32x_fma	touint
FP_DECIMAL_DIG	todec64x_div	toflt32x_mul	touintx

With changed rules for reserved identifiers, the following additions to C2x remain as these are, but these are then reserved identifiers:

<b>CHAR_WIDTH</b>	FLT128_MANT_DIG	FLT16_EPSILON	FLT32X_DIG
DEC128X_EPSILON	FLT128_MAX	FLT16_MANT_DIG	FLT32X_EPSILON
DEC128X_MANT_DIG	FLT128_MAX_10_EXP	FLT16_MAX	FLT32X_MANT_DIG
DEC128X_MAX	FLT128_MAX_EXP	FLT16_MAX_10_EXP	FLT32X_MAX
DEC128X_MAX_EXP	FLT128_MIN	FLT16_MAX_EXP	FLT32X_MAX_10_EXP
DEC128X_MIN	FLT128_MIN_10_EXP	FLT16_MIN	FLT32X_MAX_EXP
DEC128X_MIN_EXP	FLT128_MIN_EXP	FLT16_MIN_10_EXP	FLT32X_MIN
DEC128X_TRUE_MIN	FLT128_TRUE_MIN	FLT16_MIN_EXP	FLT32X_MIN_10_EXP
DEC64X_EPSILON	FLT128X_DECIMAL_DIG	FLT16_TRUE_MIN	FLT32X_MIN_EXP
DEC64X_MANT_DIG	FLT128X_DIG	FLT32_DECIMAL_DIG	FLT32X_TRUE_MIN
DEC64X_MAX	FLT128X_EPSILON	FLT32_DIG	FLT64_DECIMAL_DIG
DEC64X_MAX_EXP	FLT128X_MANT_DIG	FLT32_EPSILON	FLT64_DIG
DEC64X_MIN	FLT128X_MAX	FLT32_MANT_DIG	FLT64_EPSILON
DEC64X_MIN_EXP	FLT128X_MAX_10_EXP	FLT32_MAX	FLT64_MANT_DIG
DEC64X_TRUE_MIN	FLT128X_MAX_EXP	FLT32_MAX_10_EXP	FLT64_MAX
<b>fe_dec_getround</b>	FLT128_MIN	FLT32_MAX_EXP	FLT64_MAX_10_EXP
<b>fe_dec_setround</b>	FLT128_MIN_10_EXP	FLT32_MIN	FLT64_MAX_EXP
<b>femode_t</b>	FLT128_MIN_EXP	FLT32_MIN_10_EXP	FLT64_MIN
FLT128_DECIMAL_DIG	FLT128_TRUE_MIN	FLT32_MIN_EXP	FLT64_MIN_10_EXP
FLT128_DIG	FLT16_DECIMAL_DIG	FLT32_TRUE_MIN	FLT64_MIN_EXP
FLT128_EPSILON	FLT16_DIG	FLT32X_DECIMAL_DIG	FLT64_TRUE_MIN



FLT64X_DECIMAL_DIG	FLT64X_MAX_EXP	long_double_t	UCHAR_WIDTH
FLT64X_DIG	FLT64X_MIN	LONG_WIDTH	ULLONG_WIDTH
FLT64X_EPSILON	FLT64X_MIN_10_EXP	PTRDIFF_WIDTH	ULONG_WIDTH
FLT64X_MANT_DIG	FLT64X_MIN_EXP	SCHAR_WIDTH	USHRT_WIDTH
FLT64X_MAX	FLT64X_TRUE_MIN	SHRT_WIDTH	WCHAR_WIDTH
FLT64X_MAX_10_EXP	LLONG_WIDTH	SIZE_WIDTH	WINT_WIDTH

### 3. PROPOSED CHANGES

In the following we will run the details of our strategy and formulate questions to WG14 that can be used by them to cherry-pick those parts that seem to be suitable. For that we will generally assume the following:

- Systematic name reservations (prefix or suffix) are better than black lists of individual identifiers.
- Prefix or suffix rules that are already active for one normative context can be extended relatively easy to another normative context.
- Before being published in ISO 9899 for the first time, features can be renamed.
- New types don't need keywords, identifiers do fine.

#### 3.1. Reserve the `_t` suffix for standard types

ISO 9945 (POSIX) already reserves the `_t` suffix for types (**typedef**) in the whole library section. This is in sync with the usage in the C standard, so we think that is time to integrate that rule.

Add to 6.11 (Future language directions, general part)

Identifiers ending in `_t` may be added to represent types in any of the clauses of this document.

Add to 7.31 (Future library directions, general part)

Identifiers ending in `_t` may be added to represent **typedef** names in any of the library headers.

Adding these rules, would already reserve the **femode\_t** and **long\_double\_t** identifiers.

QUESTION 5. *Shall we add a general reservation for `_t` suffixed names to C2x as proposed in N2426?*

#### 3.2. Reserve the `_WIDTH` suffix

The `_WIDTH` suffix is now systematically used for the width of integer types. These are needed in `<math.h>` for the `(u)?fromfp(x)?` functions (renamed to `to(u)?int(x)?` below) and later for the two's complement changes for integers.

We should reserve it for the use in `<stdint.h>`. Currently, it is already reserved for those macros that start with `(U)?INT`, but it would probably be better to just reserve that suffix completely.

Add to 7.31.12 p1

Macro names ending with `_WIDTH` may be added to the macros defined in the `<stdint.h>` header.

QUESTION 6. *Shall C2x reserve the future use of the `_WIDTH` suffix to macros in `<stdint.h>` as given in N2426?*

### 3.3. Reserve the prefixes `DEC[0-9]+X?_` and `FLT[0-9]+X?_`

There are features that have been added to `<math.h>` that follow some naming scheme, but which had not previously reserved. The addition of the corresponding types and macros may happen “on the fly” when implementations add support for new optional floating types to their C libraries. Therefore we propose to add a scheme of new prefixes to `<math.h>`, one for each floating type.

Add to 7.31.12

External symbols starting with `DEC` or `FLT` followed by a sequence of digits, optionally followed by the letter `X`, then followed by an underscore and an uppercase letter may be added to the `<math.h>` header.

QUESTION 7. *Shall C2x reserve the future use of prefixes of the forms `DEC[0-9]+X?_` and `FLT[0-9]+X?_` for `<math.h>` as given in N2426?*

### 3.4. Reserve the prefixes `dec[0-9]+x?_`, `flt[0-9]+x?_`, `flt_` and `ldbl_`

To be able add new features to `<math.h>`, we will need some naming schemes that are

- regular (description by prefixes or suffixes)
- at least as expressive as the current implicit conventions

Therefore we propose to introduce several new prefixes to `<math.h>`, one for each floating type.

Add to 7.31.12

External symbols starting with `ldbl_` or `flt_` followed by a lowercase letter may be added to the `<math.h>` header. External symbols starting with `dec` or `flt` followed by a sequence of digits, optionally followed by the letter `x`, then followed by an underscore and a lowercase letter may be added to the `<math.h>` header.

QUESTION 8. *Shall C2x reserve the future use of prefixes of the forms `dec[0-9]+x?_`, `flt[0-9]+x?_`, `flt_` and `ldbl_` for `<math.h>` as given in N2426?*

### 3.5. Replace `d[0-9]+x?` suffixes by `dec[0-9]+x?_` prefixes

Having reserved these prefixes, it is easy to systematically replace all functions for decimal floating point that currently are foreseen with a `d[0-9]+x?` suffix (the `x` variant is for the types introduced in Annex N) by one with the corresponding `dec[0-9]+x?_` prefix.

*In the whole document: modify identifier variants added for C2x that end with a `d`, followed by a sequence of digits, optionally followed by an `x`, by removing that suffix and by prefixing them with `dec` the same sequence of digits and optional `x` and then by an underscore.*

For final lists (after application of all changes) of such identifiers see Section 5, below.

QUESTION 9. *Shall the identifiers that are new for C2x that have a `d[0-9]+x?` suffix be replaced by identifiers using `dec[0-9]+x?_` as given in N2426?*

### 3.6. Replace `f` and `l` suffixes by `flt_` and `ldbl_` prefixes

Because of the use of just one character as a discriminator, it is a bit more difficult to systematically replace all functions for standard floating point that currently are foreseen with a `f` or `l` suffix by one with the corresponding `flt_` and `ldbl_` prefix. This has to be done with some care, because in particular the `l` suffix can also be used for **long**, e.g.

In the whole document:

- Modify identifier variants added for C2x that end in **f** (for **float**) by removing the **f** suffix and adding a **flt\_** prefix.
- Modify identifier variants added for C2x that end in **l** (*ell*, for **long double**) by removing the **l** suffix and adding a **ldbl\_** prefix.

For final lists (after application of all changes) of such identifiers see Section 5, below.

QUESTION 10. *Shall the identifiers that are new for C2x that have a **f** or **l** suffix (for **float** or **long double**) be replaced by identifiers using **flt\_** and **ldbl\_** prefixes, respectively, as given in N2426?*

### 3.7. Replace **f[0-9]+x?** suffixes by **flt[0-9]+x?** prefixes

Annex N also adds similar functions with **f[0-9]+x?** suffixes.

*In Annex N: modify identifier variants added for C2x that end with a **f**, followed by a sequence of digits, optionally followed by an **x**, by removing that suffix and by prefixing them with **flt** the same sequence of digits and optional **x** and then by an underscore.*

For the final list (after application of all changes) of such identifiers see Section 5, below.

QUESTION 11. *Shall the identifiers in Annex N that are new for C2x that have a **f[0-9]+x?** suffix be replaced by identifiers using the **flt[0-9]+x?** prefix as given in N2426?*

### 3.8. Remove the reservation for the **cr** "prefix"

The new naming convention that reserves a **cr** prefix for “correct rounding” to 7.31.8 is not very viable, because it is in direct conflict with the addition of complex functions with a **c** prefix to `<complex.h>` in 7.31.1. In particular the identifier **crsqrt** has two possible derivations, namely as complex variant of **rsqrt** or as correct rounding variant of **sqrt**. In any case, a “some letters” prefix or postfix is not something that will reduce the problem much for future extensions, only systematic reservation will help.

Therefore we think that this paragraph should just be removed. The tools that we propose give enough leeway for future proposals to come up with naming conventions for such whole sets of features.

*Remove 7.31.8 p4 (Future library directions for `<math.h>`)*

QUESTION 12. *Shall 7.31.8 p4 be removed from C2x as proposed in N2426?*

### 3.9. Reserve more prefixes and suffixes

Many prefixes or suffixes are already present in other parts of the C standard or in other standards (as ISO 9945). Since code that is sought to be portable has to avoid them anyhow, it is much easier to add them as general rules to new parts. In particular, prefixes that are reserved for used for external symbols in any of the headers should not be used by any application.

#### 3.9.1. Reserve the **fe\_** prefix for `<fenv.h>`

The `<fenv.h>` header follows unfortunate naming conventions that are incomplete and not very consistent. In particular, it uses **fe** (without underscore), **fe\_** (with underscore), **FE\_** and **FENV\_**. Among these, only **FE\_** is systematically reserved.

We propose to add a **fe\_** prefix to the list:

Add to 7.31.4 (Future library directions for `<fenv.h>`)

External symbols starting with **fe\_** may be added to the `<fenv.h>` header.

QUESTION 13. *Shall we add a general reservation for `fe_` prefixed names for `<fenv.h>` to C2x as proposed in N2426?*

### 3.9.2. Reserve the `to` prefix for `<math.h>`

Strictly speaking, the `to` prefix is already prohibited for the use by applications for external symbols, since it is reserved for the `<ctype.h>` header. Nevertheless, the lists in 7.31 also serve to inform users what they might expect, so we propose to add the `to` prefix also to `<math.h>`:

Add `to` to the reserved prefixes in 7.31.8 p3.

QUESTION 14. *Shall we add a general reservation for `to` prefixed names to C2x to `<math.h>` as proposed in N2426?*

### 3.9.3. Change the prefix for new complex functions to `cmplx_`

The `<complex.h>` header follows unfortunate naming conventions, in particular it has functions that just add a one character prefix `c` to features provided by `math.h`. We propose to add a `cmplx_` prefix to the reservations:

Add to 7.31.4

[2 External symbols starting with `cmplx\_` may be added to the `<complex.h>` header.](#)

QUESTION 15. *Shall we add a general reservation for `cmplx_` prefixed names for `<complex.h>` to C2x as proposed in N2426?*

### 3.9.4. Remove the `c` prefixed names from future library directions

Now that we have better tools to add functions to `<complex.h>` we may as well give up the reservation of names that we will not use, anyhow.

Remove from 7.31.4

~~The function names `cacospi...`~~

QUESTION 16. *Shall 7.31.1 p1 be removed from C2x as proposed in N2426?*

### 3.10. Change the `f`, `d`, `d[0-9]+x?`, and `f[0-9]+x?` prefixes for functions rounding to narrower type

The functions for operations with rounding to narrower type that are introduced in C2x currently, use very short prefixes to indicate the type `to` which they round. This convention that has plenty of conflicts with identifiers with other purpose. In particular, a `f` as first letter is often used throughout.

We propose to use the prefixes with `to` as these are already reserved anyhow. This gives us the new prefixes `toflt_`, `todbl_`, `todec[0-9]+x?_`, and `toflt[0-9]+x?_`, respectively, for these operations. Typical renaming, if the all changes from above are applied, would then be

before	<code>&lt;math.h&gt;</code>	<code>&lt;tgmath.h&gt;</code>
<code>ffma</code>	<code>toflt_fma</code>	<code>toflt_fma</code>
<code>dfmal</code>	<code>ldbl_todbl_fma</code>	<code>todbl_fma</code>
<code>faddl</code>	<code>ldbl_toflt_add</code>	<code>toflt_add</code>
<code>d32addd64</code>	<code>dec64_todec32_add</code>	<code>todec32_add</code>
<code>d64sqrttd128</code>	<code>dec128_todec64_sqrt</code>	<code>todec64_sqrt</code>

So this leads to longer identifiers for the specialized features, but their compositions is much clearer. *E.g.* `dec128_todec64_sqrt` makes it immediately clear that this is about a square-root function that takes `_Decimal128` and returns `_Decimal64`.

*Before performing any renamings as proposed above, in 7.12.14 and in the corresponding clauses in Annex N, rewrite the newly introduced names by replacing prefixes `f`, `d`, `d[0-9]+x?`, and `f[0-9]+x` by `toflt_`, `todbl_`, `todec[0-9]+x?_`, and `toflt[0-9]+x?_`, respectively.*

*Use these new names as replacement for the old ones throughout the whole document.*

QUESTION 17. *Shall the functions that round result to narrower type (7.12.14 and Annex N) be renamed as indicated in N2426?*

### 3.11. Rename some of the remaining features

After we have tackled the naming of identifiers that observe some systematic pattern, some unfortunate choices for other singular identifiers remain, for which we can easily adapt changes that have them fit better to the general requirements.

#### 3.11.1. Rename the new floating point types

Introducing new types into C does not need syntax, therefore the introduction of keywords for the new floating point types is just overkill. We propose to remove the new keywords and to use `_t` suffixed identifiers instead (which we reserved above).

previous name	new name
<code>_DecimalN</code>	<code>decN_t</code>
<code>_FloatN</code>	<code>fltN_t</code>
<code>_DecimalNx</code>	<code>decNx_t</code>
<code>_FloatNx</code>	<code>fltNx_t</code>
<code>_DecimalN_t</code>	<code>decN_eval_t</code>
<code>_FloatN_t</code>	<code>fltN_eval_t</code>

The replacement in the main clauses is relatively straight forward. Unfortunately, Annex N complicates the view a little bit.

- In 6.2.6 (*Representations of types*), forward references, add "Predefined identifiers (6.4.2.2)"
- Remove the keywords `_Decimal32`, `_Decimal64`, `_Decimal128` from the list of keywords in 6.4.1
- Add the following sentence as a new paragraph to 6.4.2.2 (*Predefined Identifiers*):

If the implementation supports decimal floating types, for each existing type with a width of N bits, the identifier or the form `decN_t` is implicitly declared to refer to this type as if by a `typedef` that precedes the program text.

- In Annex N replace `_Decimal32_t`, `_Decimal64_t`, `_Decimal128_t` by `dec32_eval_t`, `dec64_eval_t` and `dec128_eval_t`, respectively
- In Annex N replace `_Float32_t`, `_Float64_t` by `flt32_eval_t`, and `flt64_eval_t`, respectively
- Elsewhere in the whole document replace `_Decimal32`, `_Decimal64`, `_Decimal128` by `dec32_t`, `dec64_t` and `dec128_t`, respectively

- Replace clause N.5.1 (Keywords) by

### N.5.1 Predefined identifiers

This subclause expands the set of predefined identifiers (6.4.2.2) to also include:

- `fltN_t`, where  $N$  is 16 or a positive multiple of 32
- `flt32x_t`
- `flt64x_t`
- `flt128x_t`
- `decN_t`, where  $N$  is a positive multiple of 32
- `dec64x_t`
- `dec128x_t`

Such identifiers are implicitly declared as if by a `typedef` that precedes the program text.

- Remove the decimal floating types from 6.7.2 (Type specifiers)
- Remove clause N.7 (Declarations)
- Elsewhere in Annex N replace `_DecimalN`, `_DecimalNx`, `_FloatN`, `_FloatNx`, by `decN_t`, `decNx_t`, `fltN_t`, `fltNx_t`, respectively

This change would not have much impact on implementations that already implement these types with the old keywords (as extension to C17) as the old keywords remain available for implementations.

QUESTION 18. Shall we rename the new floating types for C2x as indicated in N2426?

If WG14 instead prefers to push the whole issue of type names for the decimal floating point types to the library section, an alternative approach of integration into `<math.h>` (or a new header?) could be proposed. But this would require a bit more work and a new proposal.

#### 3.11.2. Change the `FENV_` prefix to a `FE_` prefix.

Up to now, the `FENV_` prefix was only used once, for the `FENV_ACCESS` pragma. This particular choice was and is quite unfortunate, because `<fenv.h>` explicitly reserves the `FE_` prefix. The collision between pragmas and macros is probably rare; when occurring directly in a `#pragma STDC` no macro expansion is performed. Nevertheless, macro expansion can occur for the preparation of a `_Pragma` operator, so the possibility of conflict remains.

The integration of the FP TS brought us two new pragmas, `FENV_ROUND` and `FENV_DEC_ROUND`. Other than wanting to be consistent with the choice for `FENV_ACCESS` there is no reason to have a `FENV_` prefix here. A `FE_` prefix would do equally well.

QUESTION 19. Shall we rename the pragmas `FENV_ROUND` and `FENV_DEC_ROUND` to `FE_ROUND` and `FE_DEC_ROUND` for C2x as indicated in N2426?

#### 3.11.3. Rename the new `<fenv.h>` features starting with `fe` to `fe_`

There are several new features introduced to `<fenv.h>`, and most of them have names that start with `fe`, but are not followed by an underscore. Prefix rules without an underscore separations are particular difficult to follow, they would exclude all English words starting with “fe” from the user’s name space. We think that there is not much reason to have such names for newly introduced identifiers in the first place. Using `fe_` would do just as well.

*For all newly introduced features in `<fenv.h>`, replace an initial `fe` that is not followed by an underscore by `fe_`.*

QUESTION 20. Shall we change newly introduced features for C2x in `<fenv.h>`, such that we replace an initial `fe` that is not followed by an underscore by `fe_` as indicated in N2426?

### 3.11.4. Change (u)?fromfp(x)? identifiers to to(u)?int(x)?

The prefix or name component **from** has not been used previously in the C standard, and these new identifiers do not follow any establish logic for new names. In addition, their real purpose (rounding to integer) is not apparent in the name at all. We propose to simply rename these features as in the following table.

	proposed replacement	
<b>fromfp</b>	toint	
<b>ufromfp</b>	touint	
<b>fromfpx</b>	tointx	
<b>ufromfpx</b>	touintx	
	without prefix rules	with prefix rules
<b>fromfpd</b> <i>N</i>	tointd <i>N</i>	dec <i>N</i> _toint
<b>fromfpd</b> <i>Nx</i>	tointd <i>Nx</i>	dec <i>Nx</i> _toint
<b>fromfpf</b>	tointf	flt_toint
<b>fromfpf</b> <i>N</i>	tointf <i>N</i>	flt <i>N</i> _toint
<b>fromfpf</b> <i>Nx</i>	tointf <i>Nx</i>	flt <i>Nx</i> _toint
<b>fromfpl</b>	tointl	ldbl_toint
<b>fromfpxd</b> <i>N</i>	tointxd <i>N</i>	dec <i>N</i> _tointx
<b>fromfpxd</b> <i>Nx</i>	tointxd <i>Nx</i>	dec <i>Nx</i> _tointx
<b>fromfpxf</b>	tointxf	flt_tointx
<b>fromfpxf</b> <i>N</i>	tointxf <i>N</i>	flt <i>N</i> _tointx
<b>fromfpxf</b> <i>Nx</i>	tointxf <i>Nx</i>	flt <i>Nx</i> _tointx
<b>fromfpxl</b>	tointxl	ldbl_tointx
<b>ufromfpd</b> <i>N</i>	touintd <i>N</i>	dec <i>N</i> _touint
<b>ufromfpd</b> <i>Nx</i>	touintd <i>Nx</i>	dec <i>Nx</i> _touint
<b>ufromfpf</b>	touintf	flt_touint
<b>ufromfpf</b> <i>N</i>	touintf <i>N</i>	flt <i>N</i> _touint
<b>ufromfpf</b> <i>Nx</i>	touintf <i>Nx</i>	flt <i>Nx</i> _touint
<b>ufromfpl</b>	touintl	ldbl_touint
<b>ufromfpxd</b> <i>N</i>	touintxd <i>N</i>	dec <i>N</i> _touintx
<b>ufromfpxd</b> <i>Nx</i>	touintxd <i>Nx</i>	dec <i>Nx</i> _touintx
<b>ufromfpxf</b>	touintxf	flt_touintx
<b>ufromfpxf</b> <i>N</i>	touintxf <i>N</i>	flt <i>N</i> _touintx
<b>ufromfpxf</b> <i>Nx</i>	touintxf <i>Nx</i>	flt <i>Nx</i> _touintx
<b>ufromfpxl</b>	touintxl	ldbl_touintx

The names and prefixes **toint** and **touint** have the advantage of being previously reserved (as they start with **to**) and that they reflect the purpose of the feature.

QUESTION 21. *Shall we rename the (u)?fromfp(x)? functions as indicated in N2426?*

### 3.11.5. Change llogb to toint\_logb

The **llogb** functions differ from the previously known **ilogb** functions in their return type, which is **long**. But the naming is trespassing into the user name space for no good reason. Also, this function name does not really indicate the purpose, namely condensing an operation and a conversion.

	proposed replacement
<b>FP_LLOGB0</b>	FP_TOINT_LOGB0
<b>FP_LLOGBNAN</b>	FP_TOINT_LOGBNAN
<b>llogb</b>	toint_logb

	without prefix rules	with prefix rules
<b>llogbf</b>	toint_logbf	flt_toint_logb
<b>llogbl</b>	toint_logbl	ldbl_toint_logb
<b>llogbd<math>N</math></b>	toint_logbd $N$	dec $N$ _toint_logb
<b>llogbd<math>Nx</math></b>	toint_logbd $Nx$	dec $Nx$ _toint_logb
<b>llogbf<math>N</math></b>	toint_logbf $N$	flt $N$ _toint_logb
<b>llogbf<math>Nx</math></b>	toint_logbf $Nx$	flt $Nx$ _toint_logb

Again we think that a prefix **toint** would much better reflect that feature. Also note that it is not expected that other functions along the line with even wider return type will be added to the C standard; 31 bit are considered wide enough for that. So the return type itself (**long** versus **long long** or **intmax\_t**) should not be reflected in the name.

QUESTION 22. *Shall we rename the **llogb** features as indicated in N2426?*

### 3.11.6. Change **llquantexp** to **toint\_quantexp**

Similar observations as above hold for the **llquantexp** feature. We propose the following replacements:

	proposed replacement	
<b>llquantexp</b>	toint_quantexp	
	without prefix rules	with prefix rules
<b>llquantexpd<math>N</math></b>	toint_quantexpd $N$	dec $N$ _toint_quantexp
<b>llquantexpd<math>Nx</math></b>	toint_quantexpd $Nx$	dec $Nx$ _toint_quantexp

The return type itself (**long long** versus **long** or **intmax\_t**) should not be reflected in the name and **toint** as a prefix should work fine.

QUESTION 23. *Shall we rename the **llquantexp** features as indicated in N2426?*

### 3.11.7. Change **SNAN** macros to **DBL\_SNAN** and similar

**SNAN** as a macro is particularly bad. Four letter acronyms are common and having them as macros or with an addition of just one other character as in **SNANF** or **SNANL** is searching for trouble. We propose the following replacements:

	proposed replacement
<b>SNAN</b>	DBL_SNAN
<b>SNANF</b>	FLT_SNAN
<b>SNANL</b>	LDBL_SNAN
<b>SNAND<math>N</math></b>	DEC $N$ _SNAN
<b>SNAND<math>Nx</math></b>	DEC $Nx$ _SNAN
<b>SNANF<math>N</math></b>	FLT $N$ _SNAN
<b>SNANF<math>Nx</math></b>	FLT $Nx$ _SNAN

QUESTION 24. *Shall we rename the **SNAN** features as indicated in N2426?*

### 3.11.8. Change **CR\_DECIMAL\_DIG** to **FP\_DECIMAL\_DIG**

The new feature **CR\_DECIMAL\_DIG** of Annex F does not follow any naming convention, even though the header `<float.h>` to which it is appended would have such conventions. This new feature macro represents a bound for the binary floating point types, therefore we use **FP\_DECIMAL\_DIG** by borrowing the **FP\_** prefix from `<math.h>`:



	proposed replacement
<del>CR_DECIMAL_DIG</del>	FP_DECIMAL_DIG

QUESTION 25. *Shall we rename **CR\_DECIMAL\_DIG** to **FP\_DECIMAL\_DIG** as indicated in N2426?*

## 4. CURRENT LISTS OF NEW RESERVED IDENTIFIERS.

<b>CHAR_WIDTH</b>	FLT16_MAX_EXP	FLT64_MIN_EXP	acosf16	asinhf128x
<b>CR_DECIMAL_DIG</b>	FLT16_MAX	FLT64_MIN	acosf32x	asinhf128
DEC128X_EPSILON	FLT16_MIN_10_EXP	FLT64_TRUE_MIN	acosf32	asinhf16
DEC128X_MANT_DIG	FLT16_MIN_EXP	HUGE_VAL_D128X	acosf64x	asinhf32x
DEC128X_MAX_EXP	FLT16_MIN	<b>HUGE_VAL_D128</b>	acosf64	asinhf32
DEC128X_MAX	FLT16_TRUE_MIN	<b>HUGE_VAL_D32</b>	acoshd128x	asinhf64x
DEC128X_MIN_EXP	FLT32X_DECIMAL_DIG	HUGE_VAL_D64X	<b>acoshd128</b>	asinhf64
DEC128X_MIN		<b>HUGE_VAL_D64</b>	<b>acoshd32</b>	asinpid128x
DEC128X_TRUE_MIN	FLT32X_DIG	HUGE_VAL_F128X	acoshd64x	<b>asinpid128</b>
DEC64X_EPSILON	FLT32X_EPSILON	HUGE_VAL_F128	<b>acoshd64</b>	<b>asinpid32</b>
DEC64X_MANT_DIG	FLT32X_MANT_DIG	HUGE_VAL_F16	acoshf128x	asinpid64x
DEC64X_MAX_EXP	FLT32X_MAX_10_EXP	HUGE_VAL_F32X	acoshf128	<b>asinpid64</b>
DEC64X_MAX	FLT32X_MAX_EXP	HUGE_VAL_F32	acoshf16	asinpif128x
DEC64X_MIN_EXP	FLT32X_MAX	HUGE_VAL_F64X	acoshf32x	asinpif128
DEC64X_MIN	FLT32X_MIN_10_EXP	HUGE_VAL_F64	acoshf32	asinpif16
DEC64X_TRUE_MIN	FLT32X_MIN_EXP	<b>LLONG_WIDTH</b>	acoshf64x	asinpif32x
<b>FENV_DEC_ROUND</b>	FLT32X_MIN	<b>LONG_WIDTH</b>	acoshf64	asinpif32
<b>FENV_ROUND</b>	FLT32X_TRUE_MIN	<b>PTRDIFF_WIDTH</b>	acospid128x	asinpif64x
FLT128X_DECIMAL_DIG	FLT32_DECIMAL_DIG	<b>SCHAR_WIDTH</b>	<b>acospid128</b>	asinpif64
	FLT32_DIG	<b>SHRT_WIDTH</b>	<b>acospid32</b>	<b>asinpif</b>
FLT128X_DIG	FLT32_EPSILON	<b>SIZE_WIDTH</b>	acospid64x	<b>asinpil</b>
FLT128X_EPSILON	FLT32_MANT_DIG	SNAND128X	<b>acospid64</b>	<b>asinpi</b>
FLT128X_MANT_DIG	FLT32_MAX_10_EXP	<b>SNAND128</b>	acospif128x	atan2d128x
FLT128X_MAX_10_EXP	FLT32_MAX_EXP	<b>SNAND32</b>	acospif128	<b>atan2d128</b>
	FLT32_MAX	SNAND64X	acospif16	<b>atan2d32</b>
FLT128X_MAX_EXP	FLT32_MIN_10_EXP	<b>SNAND64</b>	acospif32x	atan2d64x
FLT128X_MAX	FLT32_MIN_EXP	SNANF128X	acospif32	<b>atan2d64</b>
FLT128X_MIN_10_EXP	FLT32_MIN	SNANF128	acospif64x	atan2f128x
	FLT32_TRUE_MIN	SNANF16	acospif64	atan2f128
FLT128X_MIN_EXP	FLT64X_DECIMAL_DIG	SNANF32X	<b>acospif</b>	atan2f32x
FLT128X_MIN		SNANF32	<b>acospil</b>	atan2f32
FLT128X_TRUE_MIN	FLT64X_DIG	SNANF64X	<b>acospi</b>	atan2f64x
FLT128_DECIMAL_DIG	FLT64X_EPSILON	SNANF64	asind128x	atan2f64
	FLT64X_MANT_DIG	<b>SNANF</b>	<b>asind128</b>	atan2pid128x
FLT128_DIG	FLT64X_MAX_10_EXP	<b>SNANL</b>	<b>asind32</b>	<b>atan2pid128</b>
FLT128_EPSILON	FLT64X_MAX_EXP	<b>SNAN</b>	asind64x	<b>atan2pid32</b>
FLT128_MANT_DIG	FLT64X_MAX	<b>UCHAR_WIDTH</b>	<b>asind64</b>	atan2pid64x
FLT128_MAX_10_EXP	FLT64X_MIN_10_EXP	<b>ULLONG_WIDTH</b>	asinf128x	<b>atan2pid64</b>
FLT128_MAX_EXP	FLT64X_MIN_EXP	<b>ULONG_WIDTH</b>	asinf128	atan2pif128x
FLT128_MAX	FLT64X_MIN	<b>USHRT_WIDTH</b>	asinf16	atan2pif128
FLT128_MIN_10_EXP	FLT64X_TRUE_MIN	<b>WCHAR_WIDTH</b>	asinf32x	atan2pif32x
FLT128_MIN_EXP	FLT64_DECIMAL_DIG	<b>WINT_WIDTH</b>	asinf32	atan2pif32
FLT128_MIN	FLT64_DIG	acosd128x	asinf64x	atan2pif64x
FLT128_TRUE_MIN	FLT64_EPSILON	<b>acosd128</b>	asinf64	atan2pif64
FLT16_DECIMAL_DIG	FLT64_MANT_DIG	<b>acosd32</b>	asinhd128x	<b>atan2pif</b>
FLT16_DIG	FLT64_MAX_10_EXP	acosd64x	<b>asinhd128</b>	<b>atan2pil</b>
FLT16_EPSILON	FLT64_MAX_EXP	<b>acosd64</b>	<b>asinhd32</b>	<b>atan2pi</b>
FLT16_MANT_DIG	FLT64_MAX	acosf128x	asinhd64x	atand128x
FLT16_MAX_10_EXP	FLT64_MIN_10_EXP	acosf128	<b>asinhd64</b>	<b>atand128</b>

<b>atand32</b>	cacosf32	catanhf32x	cimagf128x	<b>cosd64</b>
atand64x	cacosf64x	catanhf32	cimagf128	cosf128x
<b>atand64</b>	cacosf64	catanhf64x	cimagf32x	cosf128
atanf128x	cacoshf128x	catanhf64	cimagf32	cosf16
atanf128	cacoshf128	<b>catanpi</b>	cimagf64x	cosf32x
atanf16	cacoshf32x	cbtrd128x	cimagf64	cosf32
atanf32x	cacoshf32	<b>cbtrd128</b>	<b>clog10p1</b>	cosf64x
atanf32	cacoshf64x	<b>cbtrd32</b>	<b>clog2p1</b>	cosf64
atanf64x	cacoshf64	cbtrd64x	clogf128x	coshd128x
atanf64	<b>cacospi</b>	<b>cbtrd64</b>	clogf128	<b>coshd128</b>
atanhd128x	canonicalized128x	cbtrf128x	clogf32x	<b>coshd32</b>
<b>atanhd128</b>	<b>canonicalized128</b>	cbtrf128	clogf32	coshd64x
<b>atanhd32</b>	<b>canonicalized32</b>	cbtrf16	clogf64x	<b>coshd64</b>
atanhd64x	canonicalized64x	cbtrf32x	clogf64	coshf128x
<b>atanhd64</b>	<b>canonicalized64</b>	cbtrf32	<b>clogp1</b>	coshf128
atanhf128x	canonicalizef128x	cbtrf64x	compoundnd128x	coshf16
atanhf128	canonicalizef128	cbtrf64	<b>compoundnd128</b>	coshf32x
atanhf16	canonicalizef16	<b>ccompoundn</b>	<b>compoundnd32</b>	coshf32
atanhf32x	canonicalizef32x	ccosf128x	compoundnd64x	coshf64x
atanhf32	canonicalizef32	ccosf128	<b>compoundnd64</b>	coshf64
atanhf64x	canonicalizef64x	ccosf32x	compoundnf128x	cospid128x
atanhf64	canonicalizef64	ccosf32	compoundnf128	<b>cospid128</b>
atanpid128x	<b>canonicalizef</b>	ccosf64x	compoundnf16	<b>cospid32</b>
<b>atanpid128</b>	<b>canonicalizel</b>	ccosf64	compoundnf32x	cospid64x
<b>atanpid32</b>	<b>canonicalize</b>	ccoshf128x	compoundnf32	<b>cospid64</b>
atanpid64x	cargf128x	ccoshf128	compoundnf64x	cospif128x
<b>atanpid64</b>	cargf128	ccoshf32x	compoundnf64	cospif128
atanpif128x	cargf32x	ccoshf32	<b>compoundnf</b>	cospif16
atanpif128	cargf32	ccoshf64x	<b>compoundn1</b>	cospif32x
atanpif16	cargf64x	ccoshf64	<b>compoundn</b>	cospif32
atanpif32x	cargf64	<b>ccospi</b>	conjf128x	cospif64x
atanpif32	casinf128x	ceild128x	conjf128	cospif64
atanpif64x	casinf128	<b>ceild128</b>	conjf32x	<b>cospif</b>
atanpif64	casinf32x	<b>ceild32</b>	conjf32	<b>cospil</b>
<b>atanpif</b>	casinf32	ceild64x	conjf64x	<b>cospi</b>
<b>atanpil</b>	casinf64x	<b>ceild64</b>	conjf64	cpowf128x
<b>atanpi</b>	casinf64	ceilf128x	copysignd128x	cpowf128
CMPLXF128X	casinhf128x	ceilf128	<b>copysignd128</b>	cpowf32x
CMPLXF128	casinhf128	ceilf16	<b>copysignd32</b>	cpowf32
CMPLXF32X	casinhf32x	ceilf32x	copysignd64x	cpowf64x
CMPLXF32	casinhf32	ceilf32	<b>copysignd64</b>	cpowf64
CMPLXF64X	casinhf64x	ceilf64x	copysignf128x	<b>cpown</b>
CMPLXF64	casinhf64	ceilf64	copysignf128	<b>cpowr</b>
cabsf128x	<b>casinpi</b>	<b>cexp10m1</b>	copysignf16	cprojf128x
cabsf128	catanf128x	<b>cexp10</b>	copysignf32x	cprojf128
cabsf32x	catanf128	<b>cexp2m1</b>	copysignf32	cprojf32x
cabsf32	catanf32x	cexpf128x	copysignf64x	cprojf32
cabsf64x	catanf32	cexpf128	copysignf64	cprojf64x
cabsf64	catanf64x	cexpf32x	cosd128x	cprojf64
cacosf128x	catanf64	cexpf32	<b>cosd128</b>	<b>cracos</b>
cacosf128	catanhf128x	cexpf64x	<b>cosd32</b>	<b>cracospi</b>
cacosf32x	catanhf128	cexpf64	cosd64x	<b>cracos</b>

<code>crasinh</code>	<code>csinhf128</code>	<code>d32muld128x</code>	<code>d64xsqrt128</code>	<code>erfcf64x</code>
<code>crasinpi</code>	<code>csinhf32x</code>	<code><b>d32muld128</b></code>	<code>d64xsqrt</code>	<code>erfcf64</code>
<code>crasin</code>	<code>csinhf32</code>	<code>d32muld64x</code>	<code>d64xsubd128x</code>	<code>erfd128x</code>
<code>cratan2pi</code>	<code>csinhf64x</code>	<code><b>d32muld64</b></code>	<code>d64xsubd128</code>	<code><b>erfd128</b></code>
<code>cratan2</code>	<code>csinhf64</code>	<code><b>d32mul</b></code>	<code>d64xsub</code>	<code><b>erfd32</b></code>
<code>cratanh</code>	<code><b>csinpi</b></code>	<code>d32sqrtd128x</code>	<code><b>dadd1</b></code>	<code>erfd64x</code>
<code>cratanpi</code>	<code>csqrtf128x</code>	<code><b>d32sqrtd128</b></code>	<code><b>dadd</b></code>	<code><b>erfd64</b></code>
<code>cratan</code>	<code>csqrtf128</code>	<code>d32sqrtd64x</code>	<code><b>ddiv1</b></code>	<code>erff128x</code>
<code>crcompoundn</code>	<code>csqrtf32x</code>	<code><b>d32sqrtd64</b></code>	<code><b>ddiv</b></code>	<code>erff128</code>
<code>crcosh</code>	<code>csqrtf32</code>	<code><b>d32sqrt</b></code>	<code>decodebind128x</code>	<code>erff16</code>
<code>cracospi</code>	<code>csqrtf64x</code>	<code>d32subd128x</code>	<code><b>decodebind128</b></code>	<code>erff32x</code>
<code>cracos</code>	<code>csqrtf64</code>	<code><b>d32subd128</b></code>	<code><b>decodebind32</b></code>	<code>erff32</code>
<code>crealf128x</code>	<code>ctanf128x</code>	<code>d32subd64x</code>	<code>decodebind64x</code>	<code>erff64x</code>
<code>crealf128</code>	<code>ctanf128</code>	<code><b>d32subd64</b></code>	<code><b>decodebind64</b></code>	<code>erff64</code>
<code>crealf32x</code>	<code>ctanf32x</code>	<code><b>d32sub</b></code>	<code><b>decodebin</b></code>	<code>exp10d128x</code>
<code>crealf32</code>	<code>ctanf32</code>	<code>d64addd128x</code>	<code>decodedecd128x</code>	<code><b>exp10d128</b></code>
<code>crealf64x</code>	<code>ctanf64x</code>	<code><b>d64addd128</b></code>	<code><b>decodedecd128</b></code>	<code><b>exp10d32</b></code>
<code>crealf64</code>	<code>ctanf64</code>	<code>d64addd64x</code>	<code><b>decodedecd32</b></code>	<code>exp10d64x</code>
<code>crexp10m1</code>	<code>ctanhf128x</code>	<code><b>d64add</b></code>	<code>decodedecd64x</code>	<code><b>exp10d64</b></code>
<code>crexp10</code>	<code>ctanhf128</code>	<code>d64divd128x</code>	<code><b>decodedecd64</b></code>	<code>exp10f128x</code>
<code>crexp2m1</code>	<code>ctanhf32x</code>	<code><b>d64divd128</b></code>	<code><b>decodedecd</b></code>	<code>exp10f128</code>
<code>crexp2</code>	<code>ctanhf32</code>	<code>d64divd64x</code>	<code><b>deprecated</b></code>	<code>exp10f32x</code>
<code>crexpm1</code>	<code>ctanhf64x</code>	<code><b>d64div</b></code>	<code><b>dfma1</b></code>	<code>exp10f32</code>
<code>crexp</code>	<code>ctanhf64</code>	<code>d64fmad128x</code>	<code><b>dfma</b></code>	<code>exp10f64x</code>
<code>crhypot</code>	<code><b>ctanpi</b></code>	<code><b>d64fmad128</b></code>	<code><b>dmull</b></code>	<code>exp10f64</code>
<code>crlog10p1</code>	<code>d128addd128x</code>	<code>d64fmad64x</code>	<code><b>dmul</b></code>	<code><b>exp10f</b></code>
<code>crlog10</code>	<code>d128add</code>	<code><b>d64fma</b></code>	<code><b>dsqrt1</b></code>	<code><b>exp10l</b></code>
<code>crlog1p</code>	<code>d128divd128x</code>	<code>d64muld128x</code>	<code><b>dsqrt</b></code>	<code>exp10m1d128x</code>
<code>crlog2p1</code>	<code>d128div</code>	<code><b>d64muld128</b></code>	<code><b>dsub1</b></code>	<code><b>exp10m1d128</b></code>
<code>crlog2</code>	<code>d128fmad128x</code>	<code>d64muld64x</code>	<code><b>dsub</b></code>	<code><b>exp10m1d32</b></code>
<code>crlogp1</code>	<code>d128fma</code>	<code><b>d64mul</b></code>	<code>encodebind128x</code>	<code>exp10m1d64x</code>
<code>crlog</code>	<code>d128muld128x</code>	<code>d64sqrtd128x</code>	<code><b>encodebind128</b></code>	<code><b>exp10m1d64</b></code>
<code>crootn</code>	<code>d128mul</code>	<code><b>d64sqrtd128</b></code>	<code><b>encodebind32</b></code>	<code>exp10m1f128x</code>
<code>crpow</code>	<code>d128sqrtd128x</code>	<code>d64sqrtd64x</code>	<code>encodebind64x</code>	<code>exp10m1f128</code>
<code>crpowr</code>	<code>d128sqrt</code>	<code><b>d64sqrt</b></code>	<code><b>encodebind64</b></code>	<code>exp10m1f32x</code>
<code>crpow</code>	<code>d128subd128x</code>	<code>d64subd128x</code>	<code><b>encodebin</b></code>	<code>exp10m1f32</code>
<code>crrootn</code>	<code>d128sub</code>	<code><b>d64subd128</b></code>	<code>encodedecd128x</code>	<code>exp10m1f64x</code>
<code>crsqrt</code>	<code>d32addd128x</code>	<code>d64subd64x</code>	<code><b>encodedecd128</b></code>	<code>exp10m1f64</code>
<code>crsinh</code>	<code><b>d32addd128</b></code>	<code><b>d64sub</b></code>	<code><b>encodedecd32</b></code>	<code>exp10m1f</code>
<code>crsinpi</code>	<code>d32addd64x</code>	<code>d64xaddd128x</code>	<code>encodedecd64x</code>	<code><b>exp10m11</b></code>
<code>crsin</code>	<code><b>d32addd64</b></code>	<code>d64xaddd128</code>	<code><b>encodedecd64</b></code>	<code><b>exp10m1</b></code>
<code>crsqrt</code>	<code><b>d32add</b></code>	<code>d64xadd</code>	<code><b>encodedecd</b></code>	<code><b>exp10</b></code>
<code>crtanh</code>	<code>d32divd128x</code>	<code>d64xdivd128x</code>	<code>erfcd128x</code>	<code>exp2d128x</code>
<code>crtanpi</code>	<code><b>d32divd128</b></code>	<code>d64xdivd128</code>	<code><b>erfcd128</b></code>	<code><b>exp2d128</b></code>
<code>crtan</code>	<code>d32divd64x</code>	<code>d64xdiv</code>	<code><b>erfcd32</b></code>	<code><b>exp2d32</b></code>
<code>csinf128x</code>	<code><b>d32divd64</b></code>	<code>d64xfmad128x</code>	<code>erfcd64x</code>	<code>exp2d64x</code>
<code>csinf128</code>	<code><b>d32div</b></code>	<code>d64xfmad128</code>	<code><b>erfcd64</b></code>	<code><b>exp2d64</b></code>
<code>csinf32x</code>	<code>d32fmad128x</code>	<code>d64xfma</code>	<code>erfcf128x</code>	<code>exp2f128x</code>
<code>csinf32</code>	<code><b>d32fmad128</b></code>	<code>d64xmuld128x</code>	<code>erfcf128</code>	<code>exp2f128</code>
<code>csinf64x</code>	<code>d32fmad64x</code>	<code>d64xmuld128</code>	<code>erfcf16</code>	<code>exp2f32x</code>
<code>csinf64</code>	<code><b>d32fmad64</b></code>	<code>d64xmul</code>	<code>erfcf32x</code>	<code>exp2f32</code>
<code>csinhf128x</code>	<code><b>d32fma</b></code>	<code>d64xsqrt128x</code>	<code>erfcf32</code>	<code>exp2f64x</code>

exp2f64	f16addf32x	f32fmaf128x	f32xsubf64	fdimf128
exp2m1d128x	f16addf32	f32fmaf128	f32xsub	fdimf16
<b>exp2m1d128</b>	f16addf64x	f32fmaf32x	f64addf128x	fdimf32x
<b>exp2m1d32</b>	f16addf64	f32fmaf64x	f64addf64x	fdimf32
exp2m1d64x	f16add	f32fmaf64	f64add	fdimf64x
<b>exp2m1d64</b>	f16divf128x	f32fma	f64divf128x	fdimf64
exp2m1f128x	f16divf128	f32mulf128x	f64divf64x	<b>fdivl</b>
exp2m1f128	f16divf32x	f32mulf128	f64div	<b>fdiv</b>
exp2m1f32x	f16divf32	f32mulf32x	f64fmaf128x	<b>fe_dec_getround</b>
exp2m1f32	f16divf64x	f32mulf64x	f64fmaf64x	<b>fe_dec_setround</b>
exp2m1f64x	f16divf64	f32mulf64	f64fma	<b>fegetmode</b>
exp2m1f64	f16div	f32mul	f64mulf128x	<b>femode_t</b>
<b>exp2m1f</b>	f16fmaf128x	f32sqrtf128x	f64mulf64x	<b>fesetexcept</b>
<b>exp2m1l</b>	f16fmaf128	f32sqrtf128	f64mul	<b>fesetmode</b>
<b>exp2m1</b>	f16fmaf32x	f32sqrtf32x	f64sqrtf128x	<b>fetestexceptflag</b>
expd128x	f16fmaf32	f32sqrtf64x	f64sqrtf64x	<b>ffmal</b>
<b>expd128</b>	f16fmaf64x	f32sqrtf64	f64sqrt	<b>ffma</b>
<b>expd32</b>	f16fmaf64	f32sqrt	f64subf128x	flood128x
expd64x	f16fma	f32subf128x	f64subf64x	<b>flood128</b>
<b>expd64</b>	f16mulf128x	f32subf128	f64sub	<b>flood32</b>
expf128x	f16mulf128	f32subf32x	f64xaddf128x	floorf128x
expf128	f16mulf32x	f32subf64x	f64xadd	floorf16
expf16	f16mulf32	f32subf64	f64xdivf128x	floorf32x
expf32x	f16mulf64x	f32sub	f64xdiv	floorf32
expf32	f16mulf64	f32xaddf128x	f64xfmaf128x	fmad128x
expf64x	f16mul	f32xaddf128	f64xfma	<b>fmad128</b>
expf64	f16sqrtf128x	f32xaddf64x	f64xmulf128x	<b>fmad32</b>
expm1d128x	f16sqrtf128	f32xaddf64	f64xmulf128x	fmad64x
<b>expm1d128</b>	f16sqrtf32x	f32xadd	f64xsqrtf128x	<b>fmad64</b>
<b>expm1d32</b>	f16sqrtf32	f32xdivf128x	f64xsqrt	fmaf128x
expm1d64x	f16sqrtf64x	f32xdivf128	f64xsubf128x	fmaf128
<b>expm1d64</b>	f16sqrtf64	f32xdivf64x	f64xsub	fmaf16
expm1f128x	f16sqrt	f32xdivf64	fabsd128x	fmaf32x
expm1f128	f16subf128x	f32xdiv	<b>fabsd128</b>	fmaf32
expm1f32x	f16subf128	f32xfmaf128x	<b>fabsd32</b>	fmaf64x
expm1f32	f16subf32x	f32xfmaf128	fabsd64x	fmaf64
expm1f64x	f16subf32	f32xfmaf64x	<b>fabsd64</b>	fmaxd128x
expm1f64	f16subf64x	f32xfmaf64	fabsf128x	<b>fmaxd128</b>
f128addf128x	f16subf64	f32xfma	fabsf128	<b>fmaxd32</b>
f128add	f16sub	f32xmulf128x	fabsf16	fmaxd64x
f128divf128x	f32addf128x	f32xmulf128	fabsf32x	<b>fmaxd64</b>
f128div	f32addf128	f32xmulf64x	fabsf32	fmaxf128x
f128fmaf128x	f32addf32x	f32xmulf64	fabsf64x	fmaxf128
f128fma	f32addf64x	f32xmul	fabsf64	fmaxf16
f128mulf128x	f32addf64	f32xsqrtf128x	<b>faddl</b>	fmaxf32x
f128mul	f32add	f32xsqrtf128	<b>fadd</b>	fmaxf32
f128sqrtf128x	f32divf128x	f32xsqrtf64x	fdimd128x	fmaxf64x
f128sqrt	f32divf128	f32xsqrtf64	<b>fdimd128</b>	fmaxf64
f128subf128x	f32divf32x	f32xsqrt	<b>fdimd32</b>	fmaxmagd128x
f128sub	f32divf64x	f32xsubf128x	fdimd64x	<b>fmaxmagd128</b>
f16addf128x	f32divf64	f32xsubf128	<b>fdimd64</b>	<b>fmaxmagd32</b>
f16addf128	f32div	f32xsubf64x	fdimf128x	fmaxmagd64x

<b>fmaxmagd64</b>	frexpd128x	getpayloadf128	lgammaf16	llroundf16
fmaxmagf128x	<b>frexpd128</b>	getpayloadf16	lgammaf32x	llroundf32x
fmaxmagf128	<b>frexpd32</b>	getpayloadf32x	lgammaf32	llroundf32
fmaxmagf16	frexpd64x	getpayloadf32	lgammaf64x	llroundf64x
fmaxmagf32x	<b>frexpd64</b>	getpayloadf64x	lgammaf64	llroundf64
fmaxmagf32	frexpf128x	getpayloadf64	llogbd128x	log10d128x
fmaxmagf64x	frexpf128	<b>getpayloadf</b>	<b>llogbd128</b>	<b>log10d128</b>
fmaxmagf64	frexpf16	<b>getpayloadl</b>	<b>llogbd32</b>	<b>log10d32</b>
<b>fmaxmagf</b>	frexpf32x	<b>getpayload</b>	llogbd64x	log10d64x
<b>fmaxmagl</b>	frexpf32	hypotd128x	<b>llogbd64</b>	<b>log10d64</b>
<b>fmaxmag</b>	frexpf64x	<b>hypotd128</b>	llogbf128x	log10f128x
fmind128x	frexpf64	<b>hypotd32</b>	llogbf128	log10f128
<b>fmind128</b>	fromfpd128x	hypotd64x	llogbf16	log10f32x
<b>fmind32</b>	<b>fromfpd128</b>	<b>hypotd64</b>	llogbf32x	log10f32
fmind64x	<b>fromfpd32</b>	hypotf128x	llogbf32	log10f64x
<b>fmind64</b>	fromfpd64x	hypotf128	llogbf64x	log10f64
fminf128x	<b>fromfpd64</b>	hypotf16	llogbf64	log10p1d128x
fminf128	fromfpf128x	hypotf32x	<b>llogbf</b>	<b>log10p1d128</b>
fminf16	fromfpf128	hypotf32	<b>llogbl</b>	<b>log10p1d32</b>
fminf32x	fromfpf16	hypotf64x	<b>llogb</b>	log10p1d64x
fminf32	fromfpf32x	hypotf64	llquantexpd128x	<b>log10p1d64</b>
fminf64x	fromfpf32	ilogbd128x	<b>llquantexpd128</b>	log10p1f128x
fminf64	fromfpf64x	<b>ilogbd128</b>	<b>llquantexpd32</b>	log10p1f128
fminmagd128x	fromfpf64	<b>ilogbd32</b>	llquantexpd64x	log10p1f32x
<b>fminmagd128</b>	<b>fromfpf</b>	ilogbd64x	<b>llquantexpd64</b>	log10p1f32
<b>fminmagd32</b>	<b>fromfpl</b>	<b>ilogbd64</b>	llquantexpf128x	log10p1f64x
fminmagd64x	fromfpxd128x	ilogbf128x	llquantexpf128	log10p1f64
<b>fminmagd64</b>	<b>fromfpxd128</b>	ilogbf128	llquantexpf16	<b>log10p1f</b>
fminmagf128x	<b>fromfpxd32</b>	ilogbf16	llquantexpf32x	<b>log10p11</b>
fminmagf128	fromfpxd64x	ilogbf32x	llquantexpf32	<b>log10p1</b>
fminmagf16	<b>fromfpxd64</b>	ilogbf32	llquantexpf64x	log1pd128x
fminmagf32x	fromfpxf128x	ilogbf64x	llquantexpf64	<b>log1pd128</b>
fminmagf32	fromfpxf128	ilogbf64	<b>llquantexp</b>	<b>log1pd32</b>
fminmagf64x	fromfpxf16	ldexpd128x	llrintd128x	log1pd64x
fminmagf64	fromfpxf32x	<b>ldexpd128</b>	<b>llrintd128</b>	<b>log1pd64</b>
<b>fminmagf</b>	fromfpxf32	<b>ldexpd32</b>	<b>llrintd32</b>	log1pf128x
<b>fminmagl</b>	fromfpxf64x	ldexpd64x	llrintd64x	log1pf128
<b>fminmag</b>	fromfpxf64	<b>ldexpd64</b>	<b>llrintd64</b>	log1pf32x
fmodd128x	<b>fromfpxf</b>	ldexpf128x	llrintf128x	log1pf32
<b>fmodd128</b>	<b>fromfpxl</b>	ldexpf128	llrintf128	log1pf64x
<b>fmodd32</b>	<b>fromfpx</b>	ldexpf16	llrintf16	log1pf64
fmodd64x	<b>fromfp</b>	ldexpf32x	llrintf32x	log2d128x
<b>fmodd64</b>	<b>fsqrtl</b>	ldexpf32	llrintf32	<b>log2d128</b>
fmodf128x	<b>fsqrt</b>	ldexpf64x	llrintf64x	<b>log2d32</b>
fmodf128	<b>fsubl</b>	ldexpf64	llrintf64	log2d64x
fmodf16	<b>fsub</b>	lgammad128x	llroundd128x	<b>log2d64</b>
fmodf32x	getpayloadd128x	<b>lgammad128</b>	<b>llroundd128</b>	log2f128x
fmodf32	<b>getpayloadd128</b>	<b>lgammad32</b>	<b>llroundd32</b>	log2f128
fmodf64x	<b>getpayloadd32</b>	lgammad64x	llroundd64x	log2f32x
fmodf64	getpayloadd64x	<b>lgammad64</b>	<b>llroundd64</b>	log2f32
<b>fmull</b>	<b>getpayloadd64</b>	lgammaf128x	llroundf128x	log2f64x
<b>fmul</b>	getpayloadf128x	lgammaf128	llroundf128	log2f64

log2pd128x	long_double_t	nearbyintd32	nextupd64x	powrf
log2pd128	lrintd128x	nearbyintd64x	nextupd64	powr1
log2pd32	lrintd128	nearbyintd64	nextupf128x	powr
log2pd64x	lrintd32	nearbyintf128x	nextupf128	quantized128x
log2pd64	lrintd64x	nearbyintf128	nextupf16	quantized128
log2pf128x	lrintd64	nearbyintf16	nextupf32x	quantized32
log2pf128	lrintf128x	nearbyintf32x	nextupf32	quantized64x
log2pf32x	lrintf128	nearbyintf32	nextupf64x	quantized64
log2pf32	lrintf16	nearbyintf64x	nextupf64	quantizef128x
log2pf64x	lrintf32x	nearbyintf64	nextupf64	quantizef128
log2pf64	lrintf32	nextafterd128x	nextupf	quantizef16
log2pf	lrintf64x	nextafterd128	nextupl	quantizef32x
log2pl	lrintf64	nextafterd32	nextup	quantizef32
log2p1	lroundd128x	nextafterd64x	nodiscard	quantizef64x
logbd128x	lroundd128	nextafterd64	powd128x	quantizef64
logbd128	lroundd32	nextafterd64	powd128	quantizef64
logbd32	lroundd64x	nextafterf128x	powd32	quantize
logbd64x	lroundd64	nextafterf128	powd64x	quantumd128x
logbd64	lroundf128x	nextafterf16	powd64	quantumd128
logbf128x	lroundf128	nextafterf32x	powf128x	quantumd32
logbf128	lroundf128	nextafterf32	powf128	quantumd64x
logbf16	lroundf16	nextafterf64x	powf16	quantumd64
logbf32x	lroundf32x	nextafterf64	powf32x	quantumf128x
logbf32	lroundf32	nextdown128x	powf32	quantumf128
logbf64x	lroundf64x	nextdown128	powf64x	quantumf16
logbf64	lroundf64	nextdown32	powf64	quantumf32x
logd128x	maybe_unused	nextdown64x	pownd128x	quantumf32
logd128	modfd128	nextdown64	pownd128	quantumf64x
logd32	modfd32	nextdownf128x	pownd32	quantumf64
logd64x	modfd64x	nextdownf128	pownd64x	quantum
logd64	modfd64	nextdownf16	pownd64	remainderd128x
logf128x	modff128x	nextdownf32x	pownf128x	remainderd128
logf128	modff128	nextdownf32	pownf128	remainderd32
logf16	modff16	nextdownf64x	pownf16	remainderd64x
logf32x	modff32x	nextdownf64	pownf32x	remainderd64
logf32	modff32	nextdownf	pownf32	remainderf128x
logf64x	modff64x	nextdownl	pownf64x	remainderf128
logf64	modff64	nextdown	pownf64	remainderf16
logpd128x	modff64	nexttowardd128x	pownf	remainderf32x
logpd128	nand128x	nexttowardd128	pownl	remainderf32
logpd32	nand128	nexttowardd32	pownd	remainderf64x
logpd64x	nand32	nexttowardd64x	powrd128x	remainderf64
logpd64	nand64x	nexttowardd64	powrd128	rind128x
logpf128x	nand64	nexttowardf128x	powrd32	rind128
logpf128	nanf128x	nexttowardf128	powrd64x	rind32
logpf32x	nanf128	nexttowardf16	powrd64	rind64x
logpf32	nanf16	nexttowardf32x	powrf128x	rind64
logpf64x	nanf32x	nexttowardf32	powrf128	rinf128x
logpf64	nanf32	nexttowardf64x	powrf16	rinf128
logpf	nanf64x	nexttowardf64	powrf32x	rinf16
logpl	nanf64	nextupd128x	powrf32	rinf32x
logp1	nearbyintd128x	nextupd128	powrf64x	rinf32
logp1	nearbyintd128	nextupd32	powrf64	rinf64x

rintf64	rsqrtf32	setpayloadf32	sinpif128x	tanpif128
rootnd128x	rsqrtf64x	setpayloadf64x	sinpif128	tanpif16
<b>rootnd128</b>	rsqrtf64	setpayloadf64	sinpif16	tanpif32x
rootnd32	<b>rsqrtf</b>	<b>setpayloadf</b>	sinpif32x	tanpif32
rootnd64x	<b>rsqrtl</b>	<b>setpayloadl</b>	sinpif32	tanpif64x
<b>rootnd64</b>	<b>rsqrt</b>	setpayloadsigd128x	sinpif64x	tanpif64
rootnf128x	samequantumd128x		sinpif64	<b>tanpif</b>
rootnf128	<b>samequantumd128</b>	<b>setpayloadsigd128</b>	<b>sinpif</b>	<b>tanpil</b>
rootnf16	<b>samequantumd32</b>	<b>setpayloadsigd32</b>	<b>sinpil</b>	<b>tanpi</b>
rootnf32x	samequantumd64x	setpayloadsigd64x	<b>sinpi</b>	tgammad128x
rootnf32	<b>samequantumd64</b>	<b>setpayloadsigd64</b>	sqrtd128x	<b>tgammad128</b>
rootnf64x	samequantumf128x	setpayloadsigf128x	<b>sqrtd128</b>	<b>tgammad32</b>
rootnf64	samequantumf128		<b>sqrtd32</b>	tgammad64x
<b>rootnf</b>	samequantumf16	setpayloadsigf128	sqrtd64x	<b>tgammad64</b>
<b>rootnl</b>	samequantumf32x	setpayloadsigf16	<b>sqrtd64</b>	tgammaf128x
<b>rootn</b>	samequantumf32	setpayloadsigf32x	sqrtf128x	tgammaf128
roundd128x	samequantumf64x	setpayloadsigf32	sqrtf128	tgammaf16
<b>roundd128</b>	samequantumf64	setpayloadsigf64x	sqrtf16	tgammaf32x
<b>roundd32</b>	<b>samequantum</b>	setpayloadsigf64	sqrtf32x	tgammaf32
roundd64x	scalblnd128x	<b>setpayloadsigf</b>	sqrtf32	tgammaf64x
<b>roundd64</b>	<b>scalblnd128</b>	<b>setpayloadsigl</b>	sqrtf64x	tgammaf64
roundevend128x	<b>scalblnd32</b>	<b>setpayloadsig</b>	sqrtf64	truncd128x
<b>roundevend128</b>	scalblnd64x	<b>setpayload</b>	tand128x	<b>truncd128</b>
<b>roundevend32</b>	<b>scalblnd64</b>	sind128x	<b>tand128</b>	<b>truncd32</b>
roundevend64x	scalblnf128x	<b>sind128</b>	<b>tand32</b>	truncd64x
<b>roundevend64</b>	scalblnf128	<b>sind32</b>	tand64x	<b>truncd64</b>
roundevenf128x	scalblnf16	sind64x	<b>tand64</b>	truncf128x
roundevenf128	scalblnf32x	<b>sind64</b>	tanf128x	truncf128
roundevenf16	scalblnf32	sinf128x	tanf128	truncf16
roundevenf32x	scalblnf64x	sinf128	tanf16	truncf32x
roundevenf32	scalblnf64	sinf16	tanf32x	truncf32
roundevenf64x	scalbnd128x	sinf32x	tanf32	truncf64x
roundevenf64	<b>scalbnd128</b>	sinf32	tanf64x	truncf64
<b>roundevenf</b>	<b>scalbnd32</b>	sinf64x	tanf64	ufromfpd128x
<b>roundevenl</b>	scalbnd64x	sinf64	tanhd128x	<b>ufromfpd128</b>
<b>roundeven</b>	<b>scalbnd64</b>	sinhd128x	<b>tanhd128</b>	<b>ufromfpd32</b>
roundf128x	scalbnf128x	<b>sinhd128</b>	<b>tanhd32</b>	ufromfpd64x
roundf128	scalbnf128	<b>sinhd32</b>	tanhd64x	<b>ufromfpd64</b>
roundf16	scalbnf16	sinhd64x	<b>tanhd64</b>	ufromfpf128x
roundf32x	scalbnf32x	<b>sinhd64</b>	tanhf128x	ufromfpf128
roundf32	scalbnf32	sinhf128x	tanhf128	ufromfpf16
roundf64x	scalbnf64x	sinhf128	tanhf16	ufromfpf32x
roundf64	scalbnf64	sinhf16	tanhf32x	ufromfpf32
rsqrtd128x	setpayloadd128x	sinhf32x	tanhf32	ufromfpf64x
<b>rsqrtd128</b>	<b>setpayloadd128</b>	sinhf32	tanhf64x	ufromfpf64
<b>rsqrtd32</b>	<b>setpayloadd32</b>	sinhf64x	tanhf64	<b>ufromfpf</b>
rsqrtd64x	setpayloadd64x	sinhf64	tanpid128x	<b>ufromfp1</b>
<b>rsqrtd64</b>	<b>setpayloadd64</b>	sinpid128x	<b>tanpid128</b>	ufromfpd128x
rsqrtf128x	setpayloadf128x	<b>sinpid128</b>	<b>tanpid32</b>	<b>ufromfpd128</b>
rsqrtf128	setpayloadf128	<b>sinpid32</b>	tanpid64x	<b>ufromfpd32</b>
rsqrtf16	setpayloadf16	sinpid64x	<b>tanpid64</b>	ufromfpd64x
rsqrtf32x	setpayloadf32x	<b>sinpid64</b>	tanpif128x	<b>ufromfpd64</b>



N2426:24

Jens Gustedt

ufrompxf128x  
ufrompxf128  
ufrompxf16

ufrompxf32x  
ufrompxf32  
ufrompxf64x

ufrompxf64  
ufrompxf  
ufrompxl

ufromfpx  
ufromfp

**5. RESULTING LISTS OF NEW SYSTEMATICALLY RESERVED IDENTIFIERS.**

D128_HUGE_VAL	dec128_ilogb	dec128_todec32_div	dec128x_exp
D128X_HUGE_VAL	dec128_ldexp	dec128_todec32_fma	dec128x_exp10
D32_HUGE_VAL	dec128_lgamma	dec128_todec32_mul	dec128x_exp10m1
D64_HUGE_VAL	dec128_llrint	dec128_todec32_sqrt	dec128x_exp2
D64X_HUGE_VAL	dec128_llround	dec128_todec32_sub	dec128x_exp2m1
DBL_SNAN	dec128_log	dec128_todec64_add	dec128x_expm1
dec128_acos	dec128_log10	dec128_todec64_div	dec128x_fabs
dec128_acosh	dec128_log10p1	dec128_todec64_fma	dec128x_fdim
dec128_acospi	dec128_log1p	dec128_todec64_mul	dec128x_floor
dec128_asin	dec128_log2	dec128_todec64_sqrt	dec128x_fma
dec128_asinh	dec128_log2p1	dec128_todec64_sub	dec128x_fmax
dec128_asinpi	dec128_logb	dec128_todec64x_add	dec128x_fmaxmag
dec128_atan	dec128_logp1	dec128_todec64x_div	dec128x_fmin
dec128_atan2	dec128_lrint	dec128_todec64x_fma	dec128x_fminmag
dec128_atan2pi	dec128_lround	dec128_todec64x_mul	dec128x_fmod
dec128_atanh	dec128_modf	dec128_todec64x_sqrt	dec128x_frexp
dec128_atanpi	dec128_nan	dec128_todec64x_sub	dec128x_getpayload
dec128_canonicalize	dec128_nearbyint	dec128_toint	dec128x_hypot
dec128_cbrt	dec128_nextafter	dec128_toint_logb	dec128x_ilogb
dec128_ceil	dec128_nextdown	dec128_toint_quantexp	dec128x_ldexp
dec128_compoundn	dec128_nexttoward	dec128_tointx	dec128x_lgamma
dec128_copysign	dec128_nextup	dec128_toint	dec128x_llrint
dec128_cos	dec128_pow	dec128_tointx	dec128x_llround
dec128_cosh	dec128_pown	dec128_trunc	dec128x_log
dec128_cospi	dec128_powr	dec128x_acos	dec128x_log10
dec128_decodebin	dec128_quantize	dec128x_acosh	dec128x_log10p1
dec128_decodedec	dec128_quantum	dec128x_acospi	dec128x_log1p
dec128_encodebin	dec128_remainder	dec128x_asin	dec128x_log2
dec128_encodedec	dec128_rint	dec128x_asinh	dec128x_log2p1
dec128_erf	dec128_rootn	dec128x_asinpi	dec128x_logb
dec128_erfc	dec128_round	dec128x_atan	dec128x_logp1
dec128_exp	dec128_roundeven	dec128x_atan2	dec128x_lrint
dec128_exp10	dec128_rsqrtd	dec128x_atan2pi	dec128x_lround
dec128_exp10m1	dec128_samequantum	dec128x_atanh	dec128x_modf
dec128_exp2	dec128_scalbln	dec128x_atanpi	dec128x_nan
dec128_exp2m1	dec128_scalbn	dec128x_canonicalize	dec128x_nearbyint
dec128_expm1	dec128_setpayload	dec128x_cbrt	dec128x_nextafter
dec128_fabs	dec128_setpayloadsig	dec128x_ceil	dec128x_nextdown
dec128_fdim	dec128_sin	dec128x_compoundn	dec128x_nexttoward
dec128_floor	dec128_sinh	dec128x_copysign	dec128x_nextup
dec128_fma	dec128_sinpi	dec128x_cos	dec128x_pown
dec128_fmax	DEC128_SNAN	dec128x_cosh	dec128x_powr
dec128_fmaxmag	dec128_sqrt	dec128x_cospi	dec128x_quantize
dec128_fmin	dec128_t	dec128x_decodebin	dec128x_quantum
dec128_fminmag	dec128_tan	dec128x_decodedec	dec128x_remainder
dec128_fmod	dec128_tanh	dec128x_encodebin	dec128x_rint
dec128_frexp	dec128_tanpi	dec128x_encodedec	dec128x_rootn
dec128_getpayload	dec128_tgamma	dec128x_erf	dec128x_round
dec128_hypot	dec128_todec32_add	dec128x_erfc	

dec128x_roundeven	dec32_asinh	dec32_lrint	dec64_atanh
dec128x_rsqrt	dec32_asinpi	dec32_lround	dec64_atanpi
dec128x_samequantum	dec32_atan	dec32_modf	dec64_canonicalize
dec128x_scalbln	dec32_atan2	dec32_nan	dec64_cbrt
dec128x_scalbn	dec32_atan2pi	dec32_nearbyint	dec64_ceil
dec128x_setpayload	dec32_atanh	dec32_nextafter	dec64_compoundn
dec128x_setpayloadsig	dec32_atanpi	dec32_nextdown	dec64_copysign
dec128x_sin	dec32_canonicalize	dec32_nexttoward	dec64_cos
dec128x_sinh	dec32_cbrt	dec32_nextup	dec64_cosh
dec128x_sinpi	dec32_ceil	dec32_pow	dec64_cospi
DEC128X_SNAN	dec32_compoundn	dec32_pown	dec64_decodebin
dec128x_sqrt	dec32_copysign	dec32_powr	dec64_decodedec
dec128x_t	dec32_cos	dec32_quantize	dec64_encodebin
dec128x_tan	dec32_cosh	dec32_quantum	dec64_encodedec
dec128x_tanh	dec32_cospi	dec32_remainder	dec64_erf
dec128x_tanpi	dec32_decodebin	dec32_rint	dec64_erfc
dec128x_tgamma	dec32_decodedec	dec32_rootn	dec64_exp
dec128x_todec128_add	dec32_encodebin	dec32_round	dec64_exp10
dec128x_todec128_div	dec32_encodedec	dec32_roundeven	dec64_exp10m1
dec128x_todec128_fma	dec32_erf	dec32_rsqrt	dec64_exp2
dec128x_todec128_mul	dec32_erfc	dec32_samequantum	dec64_exp2m1
dec128x_todec128_sqrt	dec32_exp	dec32_scalbln	dec64_expm1
dec128x_todec128_sub	dec32_exp10	dec32_scalbn	dec64_fabs
dec128x_todec32_add	dec32_exp10m1	dec32_setpayload	dec64_fdim
dec128x_todec32_div	dec32_exp2	dec32_setpayloadsig	dec64_fma
dec128x_todec32_fma	dec32_exp2m1	dec32_sin	dec64_fmax
dec128x_todec32_mul	dec32_expm1	dec32_sinh	dec64_fmaxmag
dec128x_todec32_sqrt	dec32_fabs	dec32_sinpi	dec64_fmin
dec128x_todec32_sub	dec32_fdim	DEC32_SNAN	dec64_fminmag
dec128x_todec64_add	dec32_floor	dec32_sqrt	dec64_fmod
dec128x_todec64_div	dec32_fma	dec32_t	dec64_frexp
dec128x_todec64_fma	dec32_fmax	dec32_tan	dec64_getpayload
dec128x_todec64_mul	dec32_fmaxmag	dec32_tanh	dec64_hypot
dec128x_todec64_sqrt	dec32_fmin	dec32_tanpi	dec64_ilogb
dec128x_todec64_sub	dec32_fminmag	dec32_tgamma	dec64_ldexp
dec128x_todec64x_add	dec32_fmod	dec32_toint	dec64_lgamma
dec128x_todec64x_div	dec32_frexp	dec32_toint_logb	dec64_llrint
dec128x_todec64x_fma	dec32_getpayload	dec32_toint_quantexp	dec64_llround
dec128x_todec64x_mul	dec32_hypot	dec32_tointx	dec64_log
dec128x_todec64x_sqrt	dec32_ilogb	dec32_toint	dec64_log10
dec128x_todec64x_sub	dec32_ldexp	dec32_tointx	dec64_log10p1
dec128x_toint	dec32_lgamma	dec32_trunc	dec64_log1p
dec128x_toint_logb	dec32_llrint	dec32x_t	dec64_log2
dec128x_toint_quantexp	dec32_llround	dec64_acos	dec64_log2p1
dec128x_tointx	dec32_log	dec64_acosh	dec64_logb
dec128x_toint	dec32_log10	dec64_acospi	dec64_logp1
dec128x_tointx	dec32_log10p1	dec64_asin	dec64_lrint
dec128x_trunc	dec32_log1p	dec64_asinh	dec64_lround
dec32_acos	dec32_log2	dec64_asinpi	dec64_modf
dec32_acosh	dec32_log2p1	dec64_atan	dec64_nan
dec32_acospi	dec32_logb	dec64_atan2	dec64_nearbyint
dec32_asin	dec32_logp1	dec64_atan2pi	dec64_nextafter

dec64_nextdown	dec64x_atanpi	dec64x_nexttoward	F32_HUGE_VAL
dec64_nexttoward	dec64x_canonicalize	dec64x_nextup	F32X_HUGE_VAL
dec64_nextup	dec64x_cbrt	dec64x_pow	F64_HUGE_VAL
dec64_pow	dec64x_ceil	dec64x_pown	F64X_HUGE_VAL
dec64_pown	dec64x_compoundn	dec64x_powr	FE_DEC_ROUND
dec64_powr	dec64x_copysign	dec64x_quantize	fe_getmode
dec64_quantize	dec64x_cos	dec64x_quantum	FE_ROUND
dec64_quantum	dec64x_cosh	dec64x_remainder	fe_setexcept
dec64_remainder	dec64x_cospi	dec64x_rint	fe_setmode
dec64_rint	dec64x_decodebin	dec64x_rootn	fe_testexceptflag
dec64_rootn	dec64x_decodedec	dec64x_round	float128_t
dec64_round	dec64x_encodebin	dec64x_roundeven	float32_t
dec64_roundeven	dec64x_encodedec	dec64x_rsqrtd	float64_t
dec64_rsqrtd	dec64x_erf	dec64x_samequantum	flt128_acos
dec64_samequantum	dec64x_erfc	dec64x_scalbln	flt128_acosh
dec64_scalbln	dec64x_exp	dec64x_scalbn	flt128_acospi
dec64_scalbn	dec64x_exp10	dec64x_setpayload	flt128_asin
dec64_setpayload	dec64x_exp10m1	dec64x_setpayloadsig	flt128_asinh
dec64_setpayloadsig	dec64x_exp2	dec64x_sin	flt128_asinpi
dec64_sin	dec64x_exp2m1	dec64x_sinh	flt128_atan
dec64_sinh	dec64x_expm1	dec64x_sinpi	flt128_atan2
dec64_sinpi	dec64x_fabs	DEC64X_SNAN	flt128_atan2pi
DEC64_SNAN	dec64x_fdim	dec64x_sqrt	flt128_atanh
dec64_sqrt	dec64x_fma	dec64x_t	flt128_atanpi
dec64_t	dec64x_fmax	dec64x_tan	flt128_canonicalize
dec64_tan	dec64x_fmaxmag	dec64x_tanh	flt128_cbrt
dec64_tanh	dec64x_fmin	dec64x_tanpi	flt128_ceil
dec64_tanpi	dec64x_fminmag	dec64x_tgamma	flt128_compoundn
dec64_tgamma	dec64x_fmod	dec64x_todec32_add	flt128_copysign
dec64_todec32_add	dec64x_frexpd	dec64x_todec32_div	flt128_cos
dec64_todec32_div	dec64x_getpayload	dec64x_todec32_fma	flt128_cosh
dec64_todec32_fma	dec64x_hypot	dec64x_todec32_mul	flt128_cospi
dec64_todec32_mul	dec64x_ilogb	dec64x_todec32_sqrt	flt128_decodebin
dec64_todec32_sqrt	dec64x_ldexp	dec64x_todec32_sub	flt128_decodedec
dec64_todec32_sub	dec64x_lgamma	dec64x_todec64_add	flt128_encodebin
dec64_toint	dec64x_llrint	dec64x_todec64_div	flt128_encodedec
dec64_toint_logb	dec64x_llround	dec64x_todec64_fma	flt128_erf
dec64_toint_quantexp	dec64x_log	dec64x_todec64_mul	flt128_erfc
dec64_tointx	dec64x_log10	dec64x_todec64_sqrt	flt128_exp
dec64_touint	dec64x_log10p1	dec64x_todec64_sub	flt128_exp10
dec64_touintx	dec64x_log1p	dec64x_toint	flt128_exp10m1
dec64_trunc	dec64x_log2	dec64x_toint_logb	flt128_exp2
dec64x_acos	dec64x_log2p1	dec64x_toint_quantexp	flt128_exp2m1
dec64x_acosh	dec64x_logb	dec64x_tointx	flt128_expm1
dec64x_acospi	dec64x_logp1	dec64x_touint	flt128_fabs
dec64x_asin	dec64x_lrint	dec64x_touintx	flt128_fdim
dec64x_asinh	dec64x_lround	dec64x_trunc	flt128_fma
dec64x_asinpi	dec64x_modf	decimal128_t	flt128_fmax
dec64x_atan	dec64x_nan	decimal32_t	flt128_fmaxmag
dec64x_atan2	dec64x_nearbyint	decimal64_t	flt128_fmin
dec64x_atan2pi	dec64x_nextafter	F128_HUGE_VAL	flt128_fminmag
dec64x_atanh	dec64x_nextdown	F128X_HUGE_VAL	flt128_fmod

flt128_frexp	flt128_toflt32_div	flt128x_fma	flt128x_t
flt128_getpayload	flt128_toflt32_fma	flt128x_fmax	flt128x_tan
flt128_hypot	flt128_toflt32_mul	flt128x_fmaxmag	flt128x_tanh
flt128_ilogb	flt128_toflt32_sqrt	flt128x_fmin	flt128x_tanpi
flt128_ldexp	flt128_toflt32_sub	flt128x_fminmag	flt128x_tgamma
flt128_lgamma	flt128_toflt32x_add	flt128x_fmod	flt128x_toflt128_add
flt128_llrint	flt128_toflt32x_div	flt128x_frexp	flt128x_toflt128_div
flt128_llround	flt128_toflt32x_fma	flt128x_getpayload	flt128x_toflt128_fma
flt128_log	flt128_toflt32x_mul	flt128x_hypot	flt128x_toflt128_mul
flt128_log10	flt128_toflt32x_sqrt	flt128x_ilogb	flt128x_toflt128_sqrt
flt128_log10p1	flt128_toflt32x_sub	flt128x_ldexp	flt128x_toflt128_sub
flt128_log1p	flt128_toint	flt128x_lgamma	flt128x_toflt32_add
flt128_log2	flt128_toint_logb	flt128x_llrint	flt128x_toflt32_div
flt128_log2p1	flt128_toint_quantexp	flt128x_llround	flt128x_toflt32_fma
flt128_logb	flt128_tointx	flt128x_log	flt128x_toflt32_mul
flt128_logp1	flt128_touint	flt128x_log10	flt128x_toflt32_sqrt
flt128_lrint	flt128_touintx	flt128x_log10p1	flt128x_toflt32_sub
flt128_lround	flt128_trunc	flt128x_log1p	flt128x_toflt32x_add
flt128_modf	flt128x_acos	flt128x_log2	flt128x_toflt32x_div
flt128_nan	flt128x_acosh	flt128x_log2p1	flt128x_toflt32x_fma
flt128_nearbyint	flt128x_acospi	flt128x_logb	flt128x_toflt32x_mul
flt128_nextafter	flt128x_asin	flt128x_logp1	flt128x_toflt32x_sqrt
flt128_nextdown	flt128x_asinh	flt128x_lrrint	flt128x_toflt32x_sub
flt128_nexttoward	flt128x_asinpi	flt128x_lround	flt128x_toflt64_add
flt128_nextup	flt128x_atan	flt128x_modf	flt128x_toflt64_div
flt128_pow	flt128x_atan2	flt128x_nan	flt128x_toflt64_fma
flt128_pown	flt128x_atan2pi	flt128x_nearbyint	flt128x_toflt64_mul
flt128_powr	flt128x_atanh	flt128x_nextafter	flt128x_toflt64_sqrt
flt128_quantize	flt128x_atanpi	flt128x_nextdown	flt128x_toflt64_sub
flt128_quantum	flt128x_canonicalize	flt128x_nexttoward	flt128x_toflt64x_add
flt128_remainder	flt128x_cbrt	flt128x_nextup	flt128x_toflt64x_div
flt128_rint	flt128x_ceil	flt128x_pow	flt128x_toflt64x_fma
flt128_rootn	flt128x_compoundn	flt128x_pown	flt128x_toflt64x_mul
flt128_round	flt128x_copysign	flt128x_powr	flt128x_toflt64x_sqrt
flt128_roun-even	flt128x_cos	flt128x_quantize	flt128x_toflt64x_sub
flt128_rsqr-t	flt128x_cosh	flt128x_quantum	flt128x_toint
flt128_samequantum	flt128x_cospi	flt128x_remainder	flt128x_toint_logb
flt128_scalbln	flt128x_decodebin	flt128x_rint	flt128x_toint_quantexp
flt128_scalbn	flt128x_decodedec	flt128x_rootn	flt128x_tointx
flt128_setpayload	flt128x_encodedec	flt128x_round	flt128x_touint
flt128_setpayloadsig	flt128x_encodedec	flt128x_roun-even	flt128x_touintx
flt128_sin	flt128x_erf	flt128x_rsqr-t	flt128x_trunc
flt128_sinh	flt128x_erfc	flt128x_samequantum	flt32_acos
flt128_sinpi	flt128x_exp	flt128x_scalbln	flt32_acosh
FLT128_SNaN	flt128x_exp10	flt128x_scalbn	flt32_acospi
flt128_sqrt	flt128x_exp10m1	flt128x_setpayload	flt32_asin
flt128_t	flt128x_exp2	flt128x_setpayloadsig	flt32_asinh
flt128_tan	flt128x_exp2m1	flt128x_sin	flt32_asinpi
flt128_tanh	flt128x_expm1	flt128x_sinh	flt32_atan
flt128_tanpi	flt128x_fabs	flt128x_sinpi	flt32_atan2
flt128_tgamma	flt128x_fdim	FLT128X_SNaN	flt32_atan2pi
flt128_toflt32_add	flt128x_floor	flt128x_sqrt	flt32_atanh

flt32_atanpi	flt32_nextdown	flt32x_cos	flt32x_quantize
flt32_canonicalize	flt32_nexttoward	flt32x_cosh	flt32x_quantum
flt32_cbrt	flt32_nextup	flt32x_cospi	flt32x_remainder
flt32_ceil	flt32_pow	flt32x_decodebin	flt32x_rint
flt32_compoundn	flt32_pown	flt32x_decodedec	flt32x_rootn
flt32_copysign	flt32_powr	flt32x_encodebin	flt32x_round
flt32_cos	flt32_quantize	flt32x_encodedec	flt32x_roun-even
flt32_cosh	flt32_quantum	flt32x_erf	flt32x_rsqr-t
flt32_cospi	flt32_remainder	flt32x_erfc	flt32x_samequantum
flt32_decodebin	flt32_rint	flt32x_exp	flt32x_scalbln
flt32_decodedec	flt32_rootn	flt32x_exp10	flt32x_scalbn
flt32_encodebin	flt32_round	flt32x_exp10m1	flt32x_setpayload
flt32_encodedec	flt32_roun-even	flt32x_exp2	flt32x_setpayloadsig
flt32_erf	flt32_rsqr-t	flt32x_exp2m1	flt32x_sin
flt32_erfc	flt32_samequantum	flt32x_expm1	flt32x_sinh
flt32_exp	flt32_scalbln	flt32x_fabs	flt32x_sinpi
flt32_exp10	flt32_scalbn	flt32x_fdim	FLT32X_SNaN
flt32_exp10m1	flt32_setpayload	flt32x_floor	flt32x_sqrt
flt32_exp2	flt32_setpayloadsig	flt32x_fma	flt32x_t
flt32_exp2m1	flt32_sin	flt32x_fmax	flt32x_tan
flt32_expm1	flt32_sinh	flt32x_fmaxmag	flt32x_tanh
flt32_fabs	flt32_sinpi	flt32x_fmin	flt32x_tanpi
flt32_fdim	FLT32_SNaN	flt32x_fminmag	flt32x_tgamma
flt32_floor	flt32_sqrt	flt32x_fmod	flt32x_toflt32_add
flt32_fma	flt32_t	flt32x_frex-p	flt32x_toflt32_div
flt32_fmax	flt32_tan	flt32x_getpayload	flt32x_toflt32_fma
flt32_fmaxmag	flt32_tanh	flt32x_hypot	flt32x_toflt32_mul
flt32_fmin	flt32_tanpi	flt32x_ilogb	flt32x_toflt32_sqrt
flt32_fminmag	flt32_tgamma	flt32x_ldexp	flt32x_toflt32_sub
flt32_fmod	flt32_toint	flt32x_lgamma	flt32x_toint
flt32_frex-p	flt32_toint_logb	flt32x_llrint	flt32x_toint_logb
flt32_getpayload	flt32_toint_quantexp	flt32x_llround	flt32x_toint_quantexp
flt32_hypot	flt32_tointx	flt32x_log	flt32x_tointx
flt32_ilogb	flt32_touint	flt32x_log10	flt32x_touint
flt32_ldexp	flt32_touintx	flt32x_log10p1	flt32x_touintx
flt32_lgamma	flt32_trunc	flt32x_log1p	flt32x_trunc
flt32_llrint	flt32x_acos	flt32x_log2	flt64_acos
flt32_llround	flt32x_acosh	flt32x_log2p1	flt64_acosh
flt32_log	flt32x_acospi	flt32x_logb	flt64_acospi
flt32_log10	flt32x_asin	flt32x_logp1	flt64_asin
flt32_log10p1	flt32x_asinh	flt32x_lrint	flt64_asinh
flt32_log1p	flt32x_asinpi	flt32x_lround	flt64_asinpi
flt32_log2	flt32x_atan	flt32x_modf	flt64_atan
flt32_log2p1	flt32x_atan2	flt32x_nan	flt64_atan2
flt32_logb	flt32x_atan2pi	flt32x_nearbyint	flt64_atan2pi
flt32_logp1	flt32x_atanh	flt32x_nextafter	flt64_atanh
flt32_lrint	flt32x_atanpi	flt32x_nextdown	flt64_atanpi
flt32_lround	flt32x_canonicalize	flt32x_nexttoward	flt64_canonicalize
flt32_modf	flt32x_cbrt	flt32x_nextup	flt64_cbrt
flt32_nan	flt32x_ceil	flt32x_pow	flt64_ceil
flt32_nearbyint	flt32x_compoundn	flt32x_pown	flt64_compoundn
flt32_nextafter	flt32x_copysign	flt32x_powr	flt64_copysign

flt64_cos	flt64_quantum	flt64x_canonicalize	flt64x_nextup
flt64_cosh	flt64_remainder	flt64x_cbrt	flt64x_pow
flt64_cospi	flt64_rint	flt64x_ceil	flt64x_pown
flt64_decodebin	flt64_rootn	flt64x_compoundn	flt64x_powr
flt64_decodedec	flt64_round	flt64x_copysign	flt64x_quantize
flt64_encodebin	flt64_rounleven	flt64x_cos	flt64x_quantum
flt64_encodedec	flt64_rsqrtd	flt64x_cosh	flt64x_remainder
flt64_erf	flt64_samequantum	flt64x_cospi	flt64x_rint
flt64_erfc	flt64_scalbln	flt64x_decodebin	flt64x_rootn
flt64_exp	flt64_scalbn	flt64x_decodedec	flt64x_round
flt64_exp10	flt64_setpayload	flt64x_encodebin	flt64x_rounleven
flt64_exp10m1	flt64_setpayloadsig	flt64x_encodedec	flt64x_rsqrtd
flt64_exp2	flt64_sin	flt64x_erf	flt64x_samequantum
flt64_exp2m1	flt64_sinh	flt64x_erfc	flt64x_scalbln
flt64_expm1	flt64_sinpi	flt64x_exp	flt64x_scalbn
flt64_fabs	FLT64_SNaN	flt64x_exp10	flt64x_setpayload
flt64_fdim	flt64_sqrt	flt64x_exp10m1	flt64x_setpayloadsig
flt64_fma	flt64_t	flt64x_exp2	flt64x_sin
flt64_fmax	flt64_tan	flt64x_exp2m1	flt64x_sinh
flt64_fmaxmag	flt64_tanh	flt64x_expm1	flt64x_sinpi
flt64_fmin	flt64_tanpi	flt64x_fabs	FLT64X_SNaN
flt64_fminmag	flt64_tgamma	flt64x_fdim	flt64x_sqrt
flt64_fmod	flt64_toflt32_add	flt64x_fma	flt64x_t
flt64_frexp	flt64_toflt32_div	flt64x_fmax	flt64x_tan
flt64_getpayload	flt64_toflt32_fma	flt64x_fmaxmag	flt64x_tanh
flt64_hypot	flt64_toflt32_mul	flt64x_fmin	flt64x_tanpi
flt64_ilogb	flt64_toflt32_sqrt	flt64x_fminmag	flt64x_tgamma
flt64_ldexp	flt64_toflt32_sub	flt64x_fmod	flt64x_toflt32_add
flt64_lgamma	flt64_toflt32x_add	flt64x_frexp	flt64x_toflt32_div
flt64_llrint	flt64_toflt32x_div	flt64x_getpayload	flt64x_toflt32_fma
flt64_llround	flt64_toflt32x_fma	flt64x_hypot	flt64x_toflt32_mul
flt64_log	flt64_toflt32x_mul	flt64x_ilogb	flt64x_toflt32_sqrt
flt64_log10	flt64_toflt32x_sqrt	flt64x_ldexp	flt64x_toflt32_sub
flt64_log10p1	flt64_toflt32x_sub	flt64x_lgamma	flt64x_toflt32x_add
flt64_log1p	flt64_toint	flt64x_llrint	flt64x_toflt32x_div
flt64_log2	flt64_toint_logb	flt64x_llround	flt64x_toflt32x_fma
flt64_log2p1	flt64_toint_quantexp	flt64x_log	flt64x_toflt32x_mul
flt64_logb	flt64_tointx	flt64x_log10	flt64x_toflt32x_sqrt
flt64_logp1	flt64_tointx	flt64x_log10p1	flt64x_toflt32x_sub
flt64_lrint	flt64_tointx	flt64x_log1p	flt64x_toflt64_add
flt64_lround	flt64_trunc	flt64x_log2	flt64x_toflt64_div
flt64_modf	flt64x_acos	flt64x_log2p1	flt64x_toflt64_fma
flt64_nan	flt64x_acosh	flt64x_logb	flt64x_toflt64_mul
flt64_nearbyint	flt64x_acospi	flt64x_logp1	flt64x_toflt64_sqrt
flt64_nextafter	flt64x_asin	flt64x_lrint	flt64x_toflt64_sub
flt64_nextdown	flt64x_asinh	flt64x_lround	flt64x_toint
flt64_nexttoward	flt64x_asinpi	flt64x_modf	flt64x_toint_logb
flt64_nextup	flt64x_atan	flt64x_nan	flt64x_toint_quantexp
flt64_pow	flt64x_atan2	flt64x_nearbyint	flt64x_tointx
flt64_pown	flt64x_atan2pi	flt64x_nextafter	flt64x_tointx
flt64_powr	flt64x_atanh	flt64x_nextdown	flt64x_tointx
flt64_quantize	flt64x_atanpi	flt64x_nexttoward	flt64x_trunc

flt_acospi	ldbl_canonicalize	ldbl_tointx	toflt128_sub
flt_asinpi	ldbl_compoundn	ldbl_touint	toflt32_add
flt_atan2pi	ldbl_cospi	ldbl_touintx	toflt32_div
flt_atanpi	ldbl_exp10	todbl_add	toflt32_fma
flt_canonicalize	ldbl_exp10m1	todbl_div	toflt32_mul
flt_compoundn	ldbl_exp2m1	todbl_fma	toflt32_sqrt
flt_cospi	ldbl_fmaxmag	todbl_mul	toflt32_sub
flt_exp10	ldbl_fminmag	todbl_sqrt	toflt32x_add
flt_exp10m1	ldbl_getpayload	todbl_sub	toflt32x_div
flt_exp2m1	ldbl_log10p1	todec128_add	toflt32x_fma
flt_fmaxmag	ldbl_log2p1	todec128_div	toflt32x_mul
flt_fminmag	ldbl_logp1	todec128_fma	toflt32x_sqrt
flt_getpayload	ldbl_nextdown	todec128_mul	toflt32x_sub
flt_log10p1	ldbl_nextup	todec128_sqrt	toflt64_add
flt_log2p1	ldbl_pown	todec128_sub	toflt64_div
flt_logp1	ldbl_powr	todec32_add	toflt64_fma
flt_nextdown	ldbl_rootn	todec32_div	toflt64_mul
flt_nextup	ldbl_roundeven	todec32_fma	toflt64_sqrt
flt_pown	ldbl_rsqrtd	todec32_mul	toflt64_sub
flt_powr	ldbl_setpayload	todec32_sqrt	toflt64x_add
flt_rootn	ldbl_setpayloadsig	todec32_sub	toflt64x_div
flt_roundeven	ldbl_sinpi	todec64_add	toflt64x_fma
flt_rsqrtd	LDBL_SNAN	todec64_div	toflt64x_mul
flt_setpayload	ldbl_tanpi	todec64_fma	toflt64x_sqrt
flt_setpayloadsig	ldbl_todbl_add	todec64_mul	toflt64x_sub
flt_sinpi	ldbl_todbl_div	todec64_sqrt	toflt_add
FLT_SNAN	ldbl_todbl_fma	todec64_sub	toflt_div
flt_tanpi	ldbl_todbl_mul	todec64x_add	toflt_fma
flt_toint	ldbl_todbl_sqrt	todec64x_div	toflt_mul
flt_toint_logb	ldbl_todbl_sub	todec64x_fma	toflt_sqrt
flt_tointx	ldbl_toflt_add	todec64x_mul	toflt_sub
flt_touint	ldbl_toflt_div	todec64x_sqrt	toint
flt_touintx	ldbl_toflt_fma	todec64x_sub	toint_logb
FP_DECIMAL_DIG	ldbl_toflt_mul	toflt128_add	toint_quantexp
ldbl_acospi	ldbl_toflt_sqrt	toflt128_div	tointx
ldbl_asinpi	ldbl_toflt_sub	toflt128_fma	touint
ldbl_atan2pi	ldbl_toint	toflt128_mul	touintx
ldbl_atanpi	ldbl_toint_logb	toflt128_sqrt	



## 6. SCRIPTS FOR THE TRANSITION

Some implementations have already implemented the FP TS with the names that are given there, or are currently integrating them into their C library. Once the renaming as proposed here has been decided, we will provide a tool to perform it automatically in the form of a script for the POSIX tool `sed`, see below for the current version.

Another possibility is to provide a temporary fix in form of macro `#define`.

### 6.1. An sed replacement script

```

1 # no change for CHAR_WIDTH
2 s/\<CR_DECIMAL_DIG\>/FP_DECIMAL_DIG/g
3 # no change for DEC128X_EPSILON
4 # no change for DEC128X_MANT_DIG
5 # no change for DEC128X_MAX_EXP
6 # no change for DEC128X_MAX
7 # no change for DEC128X_MIN_EXP
8 # no change for DEC128X_MIN
9 # no change for DEC128X_TRUE_MIN
10 # no change for DEC64X_EPSILON
11 # no change for DEC64X_MANT_DIG
12 # no change for DEC64X_MAX_EXP
13 # no change for DEC64X_MAX
14 # no change for DEC64X_MIN_EXP
15 # no change for DEC64X_MIN
16 # no change for DEC64X_TRUE_MIN
17 s/\<FENV_DEC_ROUND\>/FE_DEC_ROUND/g
18 s/\<FENV_ROUND\>/FE_ROUND/g
19 # no change for FLT128X_DECIMAL_DIG
20 # no change for FLT128X_DIG
21 # no change for FLT128X_EPSILON
22 # no change for FLT128X_MANT_DIG
23 # no change for FLT128X_MAX_10_EXP
24 # no change for FLT128X_MAX_EXP
25 # no change for FLT128X_MAX
26 # no change for FLT128X_MIN_10_EXP
27 # no change for FLT128X_MIN_EXP
28 # no change for FLT128X_MIN
29 # no change for FLT128X_TRUE_MIN
30 # no change for FLT128_DECIMAL_DIG
31 # no change for FLT128_DIG
32 # no change for FLT128_EPSILON
33 # no change for FLT128_MANT_DIG
34 # no change for FLT128_MAX_10_EXP
35 # no change for FLT128_MAX_EXP
36 # no change for FLT128_MAX
37 # no change for FLT128_MIN_10_EXP
38 # no change for FLT128_MIN_EXP
39 # no change for FLT128_MIN
40 # no change for FLT128_TRUE_MIN
41 # no change for FLT16_DECIMAL_DIG
42 # no change for FLT16_DIG
43 # no change for FLT16_EPSILON
44 # no change for FLT16_MANT_DIG
45 # no change for FLT16_MAX_10_EXP
46 # no change for FLT16_MAX_EXP
47 # no change for FLT16_MAX
48 # no change for FLT16_MIN_10_EXP
49 # no change for FLT16_MIN_EXP
50 # no change for FLT16_MIN
51 # no change for FLT16_TRUE_MIN
52 # no change for FLT32X_DECIMAL_DIG

```

```

53 # no change for FLT32X_DIG
54 # no change for FLT32X_EPSILON
55 # no change for FLT32X_MANT_DIG
56 # no change for FLT32X_MAX_10_EXP
57 # no change for FLT32X_MAX_EXP
58 # no change for FLT32X_MAX
59 # no change for FLT32X_MIN_10_EXP
60 # no change for FLT32X_MIN_EXP
61 # no change for FLT32X_MIN
62 # no change for FLT32X_TRUE_MIN
63 # no change for FLT32_DECIMAL_DIG
64 # no change for FLT32_DIG
65 # no change for FLT32_EPSILON
66 # no change for FLT32_MANT_DIG
67 # no change for FLT32_MAX_10_EXP
68 # no change for FLT32_MAX_EXP
69 # no change for FLT32_MAX
70 # no change for FLT32_MIN_10_EXP
71 # no change for FLT32_MIN_EXP
72 # no change for FLT32_MIN
73 # no change for FLT32_TRUE_MIN
74 # no change for FLT64X_DECIMAL_DIG
75 # no change for FLT64X_DIG
76 # no change for FLT64X_EPSILON
77 # no change for FLT64X_MANT_DIG
78 # no change for FLT64X_MAX_10_EXP
79 # no change for FLT64X_MAX_EXP
80 # no change for FLT64X_MAX
81 # no change for FLT64X_MIN_10_EXP
82 # no change for FLT64X_MIN_EXP
83 # no change for FLT64X_MIN
84 # no change for FLT64X_TRUE_MIN
85 # no change for FLT64_DECIMAL_DIG
86 # no change for FLT64_DIG
87 # no change for FLT64_EPSILON
88 # no change for FLT64_MANT_DIG
89 # no change for FLT64_MAX_10_EXP
90 # no change for FLT64_MAX_EXP
91 # no change for FLT64_MAX
92 # no change for FLT64_MIN_10_EXP
93 # no change for FLT64_MIN_EXP
94 # no change for FLT64_MIN
95 # no change for FLT64_TRUE_MIN
96 s/\<HUGE_VAL_D128X\>/D128X_HUGE_VAL/g
97 s/\<HUGE_VAL_D128\>/D128_HUGE_VAL/g
98 s/\<HUGE_VAL_D32\>/D32_HUGE_VAL/g
99 s/\<HUGE_VAL_D64X\>/D64X_HUGE_VAL/g
100 s/\<HUGE_VAL_D64\>/D64_HUGE_VAL/g
101 s/\<HUGE_VAL_F128X\>/F128X_HUGE_VAL/g
102 s/\<HUGE_VAL_F128\>/F128_HUGE_VAL/g
103 s/\<HUGE_VAL_F16\>/F16_HUGE_VAL/g
104 s/\<HUGE_VAL_F32X\>/F32X_HUGE_VAL/g
105 s/\<HUGE_VAL_F32\>/F32_HUGE_VAL/g
106 s/\<HUGE_VAL_F64X\>/F64X_HUGE_VAL/g
107 s/\<HUGE_VAL_F64\>/F64_HUGE_VAL/g
108 # no change for LLONG_WIDTH
109 # no change for LONG_WIDTH
110 # no change for PTRDIFF_WIDTH
111 # no change for SCHAR_WIDTH
112 # no change for SHRT_WIDTH
113 # no change for SIZE_WIDTH
114 s/\<SNAND128X\>/DEC128X_SNAN/g
115 s/\<SNAND128\>/DEC128_SNAN/g

```

```

116 s/\<SNAND32\>/DEC32_SNAN/g
117 s/\<SNAND64X\>/DEC64X_SNAN/g
118 s/\<SNAND64\>/DEC64_SNAN/g
119 s/\<SNANF128X\>/FLT128X_SNAN/g
120 s/\<SNANF128\>/FLT128_SNAN/g
121 s/\<SNANF16\>/FLT16_SNAN/g
122 s/\<SNANF32X\>/FLT32X_SNAN/g
123 s/\<SNANF32\>/FLT32_SNAN/g
124 s/\<SNANF64X\>/FLT64X_SNAN/g
125 s/\<SNANF64\>/FLT64_SNAN/g
126 s/\<SNANF\>/FLT_SNAN/g
127 s/\<SNANL\>/LDBL_SNAN/g
128 s/\<SNAN\>/DBL_SNAN/g
129 # no change for UCHAR_WIDTH
130 # no change for ULLONG_WIDTH
131 # no change for ULONG_WIDTH
132 # no change for USHRT_WIDTH
133 # no change for WCHAR_WIDTH
134 # no change for WINT_WIDTH
135 s/\<acosd128x\>/dec128x_acos/g
136 s/\<acosd128\>/dec128_acos/g
137 s/\<acosd32\>/dec32_acos/g
138 s/\<acosd64x\>/dec64x_acos/g
139 s/\<acosd64\>/dec64_acos/g
140 s/\<acosf128x\>/flt128x_acos/g
141 s/\<acosf128\>/flt128_acos/g
142 s/\<acosf16\>/flt16_acos/g
143 s/\<acosf32x\>/flt32x_acos/g
144 s/\<acosf32\>/flt32_acos/g
145 s/\<acosf64x\>/flt64x_acos/g
146 s/\<acosf64\>/flt64_acos/g
147 s/\<acoshd128x\>/dec128x_acosh/g
148 s/\<acoshd128\>/dec128_acosh/g
149 s/\<acoshd32\>/dec32_acosh/g
150 s/\<acoshd64x\>/dec64x_acosh/g
151 s/\<acoshd64\>/dec64_acosh/g
152 s/\<acoshf128x\>/flt128x_acosh/g
153 s/\<acoshf128\>/flt128_acosh/g
154 s/\<acoshf16\>/flt16_acosh/g
155 s/\<acoshf32x\>/flt32x_acosh/g
156 s/\<acoshf32\>/flt32_acosh/g
157 s/\<acoshf64x\>/flt64x_acosh/g
158 s/\<acoshf64\>/flt64_acosh/g
159 s/\<acospid128x\>/dec128x_acospi/g
160 s/\<acospid128\>/dec128_acospi/g
161 s/\<acospid32\>/dec32_acospi/g
162 s/\<acospid64x\>/dec64x_acospi/g
163 s/\<acospid64\>/dec64_acospi/g
164 s/\<acospif128x\>/flt128x_acospi/g
165 s/\<acospif128\>/flt128_acospi/g
166 s/\<acospif16\>/flt16_acospi/g
167 s/\<acospif32x\>/flt32x_acospi/g
168 s/\<acospif32\>/flt32_acospi/g
169 s/\<acospif64x\>/flt64x_acospi/g
170 s/\<acospif64\>/flt64_acospi/g
171 s/\<acospif\>/flt_acospi/g
172 s/\<acospil\>/ldbl_acospi/g
173 # no change for acospi
174 s/\<asind128x\>/dec128x_asin/g
175 s/\<asind128\>/dec128_asin/g
176 s/\<asind32\>/dec32_asin/g
177 s/\<asind64x\>/dec64x_asin/g
178 s/\<asind64\>/dec64_asin/g

```

```

179 s\<asinf128x\>/flt128x_asin/g
180 s\<asinf128\>/flt128_asin/g
181 s\<asinf16\>/flt16_asin/g
182 s\<asinf32x\>/flt32x_asin/g
183 s\<asinf32\>/flt32_asin/g
184 s\<asinf64x\>/flt64x_asin/g
185 s\<asinf64\>/flt64_asin/g
186 s\<asinhd128x\>/dec128x_asinh/g
187 s\<asinhd128\>/dec128_asinh/g
188 s\<asinhd32\>/dec32_asinh/g
189 s\<asinhd64x\>/dec64x_asinh/g
190 s\<asinhd64\>/dec64_asinh/g
191 s\<asinhf128x\>/flt128x_asinh/g
192 s\<asinhf128\>/flt128_asinh/g
193 s\<asinhf16\>/flt16_asinh/g
194 s\<asinhf32x\>/flt32x_asinh/g
195 s\<asinhf32\>/flt32_asinh/g
196 s\<asinhf64x\>/flt64x_asinh/g
197 s\<asinhf64\>/flt64_asinh/g
198 s\<asinpid128x\>/dec128x_asinpi/g
199 s\<asinpid128\>/dec128_asinpi/g
200 s\<asinpid32\>/dec32_asinpi/g
201 s\<asinpid64x\>/dec64x_asinpi/g
202 s\<asinpid64\>/dec64_asinpi/g
203 s\<asinpif128x\>/flt128x_asinpi/g
204 s\<asinpif128\>/flt128_asinpi/g
205 s\<asinpif16\>/flt16_asinpi/g
206 s\<asinpif32x\>/flt32x_asinpi/g
207 s\<asinpif32\>/flt32_asinpi/g
208 s\<asinpif64x\>/flt64x_asinpi/g
209 s\<asinpif64\>/flt64_asinpi/g
210 s\<asinpif\>/flt_asinpi/g
211 s\<asinpil\>/ldbl_asinpi/g
212 # no change for asinpi
213 s\<atan2d128x\>/dec128x_atan2/g
214 s\<atan2d128\>/dec128_atan2/g
215 s\<atan2d32\>/dec32_atan2/g
216 s\<atan2d64x\>/dec64x_atan2/g
217 s\<atan2d64\>/dec64_atan2/g
218 s\<atan2f128x\>/flt128x_atan2/g
219 s\<atan2f128\>/flt128_atan2/g
220 s\<atan2f32x\>/flt32x_atan2/g
221 s\<atan2f32\>/flt32_atan2/g
222 s\<atan2f64x\>/flt64x_atan2/g
223 s\<atan2f64\>/flt64_atan2/g
224 s\<atan2pid128x\>/dec128x_atan2pi/g
225 s\<atan2pid128\>/dec128_atan2pi/g
226 s\<atan2pid32\>/dec32_atan2pi/g
227 s\<atan2pid64x\>/dec64x_atan2pi/g
228 s\<atan2pid64\>/dec64_atan2pi/g
229 s\<atan2pif128x\>/flt128x_atan2pi/g
230 s\<atan2pif128\>/flt128_atan2pi/g
231 s\<atan2pif32x\>/flt32x_atan2pi/g
232 s\<atan2pif32\>/flt32_atan2pi/g
233 s\<atan2pif64x\>/flt64x_atan2pi/g
234 s\<atan2pif64\>/flt64_atan2pi/g
235 s\<atan2pif\>/flt_atan2pi/g
236 s\<atan2pil\>/ldbl_atan2pi/g
237 # no change for atan2pi
238 s\<atand128x\>/dec128x_atan/g
239 s\<atand128\>/dec128_atan/g
240 s\<atand32\>/dec32_atan/g
241 s\<atand64x\>/dec64x_atan/g

```

```

242 s/\<atand64\>/dec64_atan/g
243 s/\<atanf128x\>/flt128x_atan/g
244 s/\<atanf128\>/flt128_atan/g
245 s/\<atanf16\>/flt16_atan/g
246 s/\<atanf32x\>/flt32x_atan/g
247 s/\<atanf32\>/flt32_atan/g
248 s/\<atanf64x\>/flt64x_atan/g
249 s/\<atanf64\>/flt64_atan/g
250 s/\<atanhd128x\>/dec128x_atanh/g
251 s/\<atanhd128\>/dec128_atanh/g
252 s/\<atanhd32\>/dec32_atanh/g
253 s/\<atanhd64x\>/dec64x_atanh/g
254 s/\<atanhd64\>/dec64_atanh/g
255 s/\<atanhf128x\>/flt128x_atanh/g
256 s/\<atanhf128\>/flt128_atanh/g
257 s/\<atanhf16\>/flt16_atanh/g
258 s/\<atanhf32x\>/flt32x_atanh/g
259 s/\<atanhf32\>/flt32_atanh/g
260 s/\<atanhf64x\>/flt64x_atanh/g
261 s/\<atanhf64\>/flt64_atanh/g
262 s/\<atanpid128x\>/dec128x_atanpi/g
263 s/\<atanpid128\>/dec128_atanpi/g
264 s/\<atanpid32\>/dec32_atanpi/g
265 s/\<atanpid64x\>/dec64x_atanpi/g
266 s/\<atanpid64\>/dec64_atanpi/g
267 s/\<atanpif128x\>/flt128x_atanpi/g
268 s/\<atanpif128\>/flt128_atanpi/g
269 s/\<atanpif16\>/flt16_atanpi/g
270 s/\<atanpif32x\>/flt32x_atanpi/g
271 s/\<atanpif32\>/flt32_atanpi/g
272 s/\<atanpif64x\>/flt64x_atanpi/g
273 s/\<atanpif64\>/flt64_atanpi/g
274 s/\<atanpif\>/flt_atanpi/g
275 s/\<atanpil\>/ldbl_atanpi/g
276 # no change for atanpi
277 # no change for Cmplxf128x
278 # no change for Cmplxf128
279 # no change for Cmplxf32x
280 # no change for Cmplxf32
281 # no change for Cmplxf64x
282 # no change for Cmplxf64
283 s/\<cabsf128x\>/flt128x_cabs/g
284 s/\<cabsf128\>/flt128_cabs/g
285 s/\<cabsf32x\>/flt32x_cabs/g
286 s/\<cabsf32\>/flt32_cabs/g
287 s/\<cabsf64x\>/flt64x_cabs/g
288 s/\<cabsf64\>/flt64_cabs/g
289 s/\<cacosf128x\>/flt128x_cacos/g
290 s/\<cacosf128\>/flt128_cacos/g
291 s/\<cacosf32x\>/flt32x_cacos/g
292 s/\<cacosf32\>/flt32_cacos/g
293 s/\<cacosf64x\>/flt64x_cacos/g
294 s/\<cacosf64\>/flt64_cacos/g
295 s/\<cacoshf128x\>/flt128x_cacosh/g
296 s/\<cacoshf128\>/flt128_cacosh/g
297 s/\<cacoshf32x\>/flt32x_cacosh/g
298 s/\<cacoshf32\>/flt32_cacosh/g
299 s/\<cacoshf64x\>/flt64x_cacosh/g
300 s/\<cacoshf64\>/flt64_cacosh/g
301 # no change for cacosp
302 s/\<canonicalized128x\>/dec128x_canonicalize/g
303 s/\<canonicalized128\>/dec128_canonicalize/g
304 s/\<canonicalized32\>/dec32_canonicalize/g

```

```

305 s\<canonicalized64x\>/dec64x_canonicalize/g
306 s\<canonicalized64\>/dec64_canonicalize/g
307 s\<canonicalizef128x\>/flt128x_canonicalize/g
308 s\<canonicalizef128\>/flt128_canonicalize/g
309 s\<canonicalizef16\>/flt16_canonicalize/g
310 s\<canonicalizef32x\>/flt32x_canonicalize/g
311 s\<canonicalizef32\>/flt32_canonicalize/g
312 s\<canonicalizef64x\>/flt64x_canonicalize/g
313 s\<canonicalizef64\>/flt64_canonicalize/g
314 s\<canonicalizef\>/flt_canonicalize/g
315 s\<canonicalizel\>/ldbl_canonicalize/g
316 # no change for canonicalize
317 s\<cargf128x\>/flt128x_carg/g
318 s\<cargf128\>/flt128_carg/g
319 s\<cargf32x\>/flt32x_carg/g
320 s\<cargf32\>/flt32_carg/g
321 s\<cargf64x\>/flt64x_carg/g
322 s\<cargf64\>/flt64_carg/g
323 s\<casinf128x\>/flt128x_casin/g
324 s\<casinf128\>/flt128_casin/g
325 s\<casinf32x\>/flt32x_casin/g
326 s\<casinf32\>/flt32_casin/g
327 s\<casinf64x\>/flt64x_casin/g
328 s\<casinf64\>/flt64_casin/g
329 s\<casinhf128x\>/flt128x_casinh/g
330 s\<casinhf128\>/flt128_casinh/g
331 s\<casinhf32x\>/flt32x_casinh/g
332 s\<casinhf32\>/flt32_casinh/g
333 s\<casinhf64x\>/flt64x_casinh/g
334 s\<casinhf64\>/flt64_casinh/g
335 # no change for casinpi
336 s\<catanf128x\>/flt128x_catan/g
337 s\<catanf128\>/flt128_catan/g
338 s\<catanf32x\>/flt32x_catan/g
339 s\<catanf32\>/flt32_catan/g
340 s\<catanf64x\>/flt64x_catan/g
341 s\<catanf64\>/flt64_catan/g
342 s\<catanhf128x\>/flt128x_catanh/g
343 s\<catanhf128\>/flt128_catanh/g
344 s\<catanhf32x\>/flt32x_catanh/g
345 s\<catanhf32\>/flt32_catanh/g
346 s\<catanhf64x\>/flt64x_catanh/g
347 s\<catanhf64\>/flt64_catanh/g
348 # no change for catanpi
349 s\<cbtrtd128x\>/dec128x_cbtrt/g
350 s\<cbtrtd128\>/dec128_cbtrt/g
351 s\<cbtrtd32\>/dec32_cbtrt/g
352 s\<cbtrtd64x\>/dec64x_cbtrt/g
353 s\<cbtrtd64\>/dec64_cbtrt/g
354 s\<cbtrtf128x\>/flt128x_cbtrt/g
355 s\<cbtrtf128\>/flt128_cbtrt/g
356 s\<cbtrtf16\>/flt16_cbtrt/g
357 s\<cbtrtf32x\>/flt32x_cbtrt/g
358 s\<cbtrtf32\>/flt32_cbtrt/g
359 s\<cbtrtf64x\>/flt64x_cbtrt/g
360 s\<cbtrtf64\>/flt64_cbtrt/g
361 # no change for ccompoundn
362 s\<ccosf128x\>/flt128x_ccos/g
363 s\<ccosf128\>/flt128_ccos/g
364 s\<ccosf32x\>/flt32x_ccos/g
365 s\<ccosf32\>/flt32_ccos/g
366 s\<ccosf64x\>/flt64x_ccos/g
367 s\<ccosf64\>/flt64_ccos/g

```

```

368 s/\<ccoshf128x\>/flt128x_ccosh/g
369 s/\<ccoshf128\>/flt128_ccosh/g
370 s/\<ccoshf32x\>/flt32x_ccosh/g
371 s/\<ccoshf32\>/flt32_ccosh/g
372 s/\<ccoshf64x\>/flt64x_ccosh/g
373 s/\<ccoshf64\>/flt64_ccosh/g
374 # no change for ccospi
375 s/\<ceild128x\>/dec128x_ceil/g
376 s/\<ceild128\>/dec128_ceil/g
377 s/\<ceild32\>/dec32_ceil/g
378 s/\<ceild64x\>/dec64x_ceil/g
379 s/\<ceild64\>/dec64_ceil/g
380 s/\<ceilf128x\>/flt128x_ceil/g
381 s/\<ceilf128\>/flt128_ceil/g
382 s/\<ceilf16\>/flt16_ceil/g
383 s/\<ceilf32x\>/flt32x_ceil/g
384 s/\<ceilf32\>/flt32_ceil/g
385 s/\<ceilf64x\>/flt64x_ceil/g
386 s/\<ceilf64\>/flt64_ceil/g
387 # no change for cexp10m1
388 # no change for cexp10
389 # no change for cexp2m1
390 s/\<cexpf128x\>/flt128x_cexp/g
391 s/\<cexpf128\>/flt128_cexp/g
392 s/\<cexpf32x\>/flt32x_cexp/g
393 s/\<cexpf32\>/flt32_cexp/g
394 s/\<cexpf64x\>/flt64x_cexp/g
395 s/\<cexpf64\>/flt64_cexp/g
396 s/\<cimagf128x\>/flt128x_cimag/g
397 s/\<cimagf128\>/flt128_cimag/g
398 s/\<cimagf32x\>/flt32x_cimag/g
399 s/\<cimagf32\>/flt32_cimag/g
400 s/\<cimagf64x\>/flt64x_cimag/g
401 s/\<cimagf64\>/flt64_cimag/g
402 # no change for clog10p1
403 # no change for clog2p1
404 s/\<clogf128x\>/flt128x_clog/g
405 s/\<clogf128\>/flt128_clog/g
406 s/\<clogf32x\>/flt32x_clog/g
407 s/\<clogf32\>/flt32_clog/g
408 s/\<clogf64x\>/flt64x_clog/g
409 s/\<clogf64\>/flt64_clog/g
410 # no change for clogp1
411 s/\<compoundnd128x\>/dec128x_compoundn/g
412 s/\<compoundnd128\>/dec128_compoundn/g
413 s/\<compoundnd32\>/dec32_compoundn/g
414 s/\<compoundnd64x\>/dec64x_compoundn/g
415 s/\<compoundnd64\>/dec64_compoundn/g
416 s/\<compoundnf128x\>/flt128x_compoundn/g
417 s/\<compoundnf128\>/flt128_compoundn/g
418 s/\<compoundnf16\>/flt16_compoundn/g
419 s/\<compoundnf32x\>/flt32x_compoundn/g
420 s/\<compoundnf32\>/flt32_compoundn/g
421 s/\<compoundnf64x\>/flt64x_compoundn/g
422 s/\<compoundnf64\>/flt64_compoundn/g
423 s/\<compoundnf\>/flt_compoundn/g
424 s/\<compoundn1\>/ldbl_compoundn/g
425 # no change for compoundn
426 s/\<conjf128x\>/flt128x_conj/g
427 s/\<conjf128\>/flt128_conj/g
428 s/\<conjf32x\>/flt32x_conj/g
429 s/\<conjf32\>/flt32_conj/g
430 s/\<conjf64x\>/flt64x_conj/g

```

```

431 s\<conjf64\>/flt64_conj/g
432 s\<copysignd128x\>/dec128_copysign/g
433 s\<copysignd128\>/dec128_copysign/g
434 s\<copysignd32\>/dec32_copysign/g
435 s\<copysignd64x\>/dec64x_copysign/g
436 s\<copysignd64\>/dec64_copysign/g
437 s\<copysignf128x\>/flt128x_copysign/g
438 s\<copysignf128\>/flt128_copysign/g
439 s\<copysignf16\>/flt16_copysign/g
440 s\<copysignf32x\>/flt32x_copysign/g
441 s\<copysignf32\>/flt32_copysign/g
442 s\<copysignf64x\>/flt64x_copysign/g
443 s\<copysignf64\>/flt64_copysign/g
444 s\<cosd128x\>/dec128x_cos/g
445 s\<cosd128\>/dec128_cos/g
446 s\<cosd32\>/dec32_cos/g
447 s\<cosd64x\>/dec64x_cos/g
448 s\<cosd64\>/dec64_cos/g
449 s\<cosf128x\>/flt128x_cos/g
450 s\<cosf128\>/flt128_cos/g
451 s\<cosf16\>/flt16_cos/g
452 s\<cosf32x\>/flt32x_cos/g
453 s\<cosf32\>/flt32_cos/g
454 s\<cosf64x\>/flt64x_cos/g
455 s\<cosf64\>/flt64_cos/g
456 s\<coshd128x\>/dec128x_cosh/g
457 s\<coshd128\>/dec128_cosh/g
458 s\<coshd32\>/dec32_cosh/g
459 s\<coshd64x\>/dec64x_cosh/g
460 s\<coshd64\>/dec64_cosh/g
461 s\<coshf128x\>/flt128x_cosh/g
462 s\<coshf128\>/flt128_cosh/g
463 s\<coshf16\>/flt16_cosh/g
464 s\<coshf32x\>/flt32x_cosh/g
465 s\<coshf32\>/flt32_cosh/g
466 s\<coshf64x\>/flt64x_cosh/g
467 s\<coshf64\>/flt64_cosh/g
468 s\<cospid128x\>/dec128x_cospi/g
469 s\<cospid128\>/dec128_cospi/g
470 s\<cospid32\>/dec32_cospi/g
471 s\<cospid64x\>/dec64x_cospi/g
472 s\<cospid64\>/dec64_cospi/g
473 s\<cospif128x\>/flt128x_cospi/g
474 s\<cospif128\>/flt128_cospi/g
475 s\<cospif16\>/flt16_cospi/g
476 s\<cospif32x\>/flt32x_cospi/g
477 s\<cospif32\>/flt32_cospi/g
478 s\<cospif64x\>/flt64x_cospi/g
479 s\<cospif64\>/flt64_cospi/g
480 s\<cospif\>/flt_cospi/g
481 s\<cospil\>/ldbl_cospi/g
482 # no change for cospi
483 s\<cpowf128x\>/flt128x_cpow/g
484 s\<cpowf128\>/flt128_cpow/g
485 s\<cpowf32x\>/flt32x_cpow/g
486 s\<cpowf32\>/flt32_cpow/g
487 s\<cpowf64x\>/flt64x_cpow/g
488 s\<cpowf64\>/flt64_cpow/g
489 # no change for cpown
490 # no change for cpowr
491 s\<cprojf128x\>/flt128x_cproj/g
492 s\<cprojf128\>/flt128_cproj/g
493 s\<cprojf32x\>/flt32x_cproj/g

```



```

494 s/\<cprojf32\>/flt32_cproj/g
495 s/\<cprojf64x\>/flt64x_cproj/g
496 s/\<cprojf64\>/flt64_cproj/g
497 # no change for cracosh
498 # no change for cracospi
499 # no change for cracos
500 # no change for crasinh
501 # no change for crasinpi
502 # no change for crasin
503 # no change for cratan2pi
504 # no change for cratan2
505 # no change for cratanh
506 # no change for cratanpi
507 # no change for cratan
508 # no change for crcompoundn
509 # no change for crcosh
510 # no change for crcospi
511 # no change for crcos
512 s/\<crealf128x\>/flt128x_creal/g
513 s/\<crealf128\>/flt128_creal/g
514 s/\<crealf32x\>/flt32x_creal/g
515 s/\<crealf32\>/flt32_creal/g
516 s/\<crealf64x\>/flt64x_creal/g
517 s/\<crealf64\>/flt64_creal/g
518 # no change for crexp10m1
519 # no change for crexp10
520 # no change for crexp2m1
521 # no change for crexp2
522 # no change for crexpm1
523 # no change for crexp
524 # no change for crhypot
525 # no change for crlog10p1
526 # no change for crlog10
527 # no change for crlog1p
528 # no change for crlog2p1
529 # no change for crlog2
530 # no change for crlogp1
531 # no change for crlog
532 # no change for crootn
533 # no change for crpown
534 # no change for crpowr
535 # no change for crpow
536 # no change for crrootn
537 # no change for crrsqrt
538 # no change for crsinh
539 # no change for crsinpi
540 # no change for crsin
541 # no change for crsqr
542 # no change for crtanh
543 # no change for crtanpi
544 # no change for crtana
545 s/\<csinf128x\>/flt128x_csin/g
546 s/\<csinf128\>/flt128_csin/g
547 s/\<csinf32x\>/flt32x_csin/g
548 s/\<csinf32\>/flt32_csin/g
549 s/\<csinf64x\>/flt64x_csin/g
550 s/\<csinf64\>/flt64_csin/g
551 s/\<csinhf128x\>/flt128x_csinh/g
552 s/\<csinhf128\>/flt128_csinh/g
553 s/\<csinhf32x\>/flt32x_csinh/g
554 s/\<csinhf32\>/flt32_csinh/g
555 s/\<csinhf64x\>/flt64x_csinh/g
556 s/\<csinhf64\>/flt64_csinh/g

```

```

557 # no change for csinpi
558 s\^<csqrtf128x\>/flt128x_csqrt/g
559 s\^<csqrtf128\>/flt128_csqrt/g
560 s\^<csqrtf32x\>/flt32x_csqrt/g
561 s\^<csqrtf32\>/flt32_csqrt/g
562 s\^<csqrtf64x\>/flt64x_csqrt/g
563 s\^<csqrtf64\>/flt64_csqrt/g
564 s\^<ctanf128x\>/flt128x_ctan/g
565 s\^<ctanf128\>/flt128_ctan/g
566 s\^<ctanf32x\>/flt32x_ctan/g
567 s\^<ctanf32\>/flt32_ctan/g
568 s\^<ctanf64x\>/flt64x_ctan/g
569 s\^<ctanf64\>/flt64_ctan/g
570 s\^<ctanhf128x\>/flt128x_ctanh/g
571 s\^<ctanhf128\>/flt128_ctanh/g
572 s\^<ctanhf32x\>/flt32x_ctanh/g
573 s\^<ctanhf32\>/flt32_ctanh/g
574 s\^<ctanhf64x\>/flt64x_ctanh/g
575 s\^<ctanhf64\>/flt64_ctanh/g
576 # no change for ctanpi
577 s\^<d128add128x\>/dec128x_todec128_add/g
578 s\^<d128add\>/todec128_add/g
579 s\^<d128divd128x\>/dec128x_todec128_div/g
580 s\^<d128div\>/todec128_div/g
581 s\^<d128fmad128x\>/dec128x_todec128_fma/g
582 s\^<d128fma\>/todec128_fma/g
583 s\^<d128muld128x\>/dec128x_todec128_mul/g
584 s\^<d128mul\>/todec128_mul/g
585 s\^<d128sqrtd128x\>/dec128x_todec128_sqrt/g
586 s\^<d128sqrt\>/todec128_sqrt/g
587 s\^<d128subd128x\>/dec128x_todec128_sub/g
588 s\^<d128sub\>/todec128_sub/g
589 s\^<d32addd128x\>/dec128x_todec32_add/g
590 s\^<d32addd128\>/dec128_todec32_add/g
591 s\^<d32addd64x\>/dec64x_todec32_add/g
592 s\^<d32addd64\>/dec64_todec32_add/g
593 s\^<d32add\>/todec32_add/g
594 s\^<d32divd128x\>/dec128x_todec32_div/g
595 s\^<d32divd128\>/dec128_todec32_div/g
596 s\^<d32divd64x\>/dec64x_todec32_div/g
597 s\^<d32divd64\>/dec64_todec32_div/g
598 s\^<d32div\>/todec32_div/g
599 s\^<d32fmad128x\>/dec128x_todec32_fma/g
600 s\^<d32fmad128\>/dec128_todec32_fma/g
601 s\^<d32fmad64x\>/dec64x_todec32_fma/g
602 s\^<d32fmad64\>/dec64_todec32_fma/g
603 s\^<d32fma\>/todec32_fma/g
604 s\^<d32muld128x\>/dec128x_todec32_mul/g
605 s\^<d32muld128\>/dec128_todec32_mul/g
606 s\^<d32muld64x\>/dec64x_todec32_mul/g
607 s\^<d32muld64\>/dec64_todec32_mul/g
608 s\^<d32mul\>/todec32_mul/g
609 s\^<d32sqrtd128x\>/dec128x_todec32_sqrt/g
610 s\^<d32sqrtd128\>/dec128_todec32_sqrt/g
611 s\^<d32sqrtd64x\>/dec64x_todec32_sqrt/g
612 s\^<d32sqrtd64\>/dec64_todec32_sqrt/g
613 s\^<d32sqrt\>/todec32_sqrt/g
614 s\^<d32subd128x\>/dec128x_todec32_sub/g
615 s\^<d32subd128\>/dec128_todec32_sub/g
616 s\^<d32subd64x\>/dec64x_todec32_sub/g
617 s\^<d32subd64\>/dec64_todec32_sub/g
618 s\^<d32sub\>/todec32_sub/g
619 s\^<d64addd128x\>/dec128x_todec64_add/g

```

```

620 s/\<d64add128\>/dec128_todec64_add/g
621 s/\<d64add64x\>/dec64x_todec64_add/g
622 s/\<d64add\>/todec64_add/g
623 s/\<d64div128x\>/dec128x_todec64_div/g
624 s/\<d64div128\>/dec128_todec64_div/g
625 s/\<d64div64x\>/dec64x_todec64_div/g
626 s/\<d64div\>/todec64_div/g
627 s/\<d64fmad128x\>/dec128x_todec64_fma/g
628 s/\<d64fmad128\>/dec128_todec64_fma/g
629 s/\<d64fmad64x\>/dec64x_todec64_fma/g
630 s/\<d64fma\>/todec64_fma/g
631 s/\<d64muld128x\>/dec128x_todec64_mul/g
632 s/\<d64muld128\>/dec128_todec64_mul/g
633 s/\<d64muld64x\>/dec64x_todec64_mul/g
634 s/\<d64mul\>/todec64_mul/g
635 s/\<d64sqrtd128x\>/dec128x_todec64_sqrt/g
636 s/\<d64sqrtd128\>/dec128_todec64_sqrt/g
637 s/\<d64sqrtd64x\>/dec64x_todec64_sqrt/g
638 s/\<d64sqrt\>/todec64_sqrt/g
639 s/\<d64subd128x\>/dec128x_todec64_sub/g
640 s/\<d64subd128\>/dec128_todec64_sub/g
641 s/\<d64subd64x\>/dec64x_todec64_sub/g
642 s/\<d64sub\>/todec64_sub/g
643 s/\<d64xadd128x\>/dec128x_todec64x_add/g
644 s/\<d64xadd128\>/dec128_todec64x_add/g
645 s/\<d64xadd\>/todec64x_add/g
646 s/\<d64xdivd128x\>/dec128x_todec64x_div/g
647 s/\<d64xdivd128\>/dec128_todec64x_div/g
648 s/\<d64xdiv\>/todec64x_div/g
649 s/\<d64xfmad128x\>/dec128x_todec64x_fma/g
650 s/\<d64xfmad128\>/dec128_todec64x_fma/g
651 s/\<d64xfma\>/todec64x_fma/g
652 s/\<d64xmuld128x\>/dec128x_todec64x_mul/g
653 s/\<d64xmuld128\>/dec128_todec64x_mul/g
654 s/\<d64xmul\>/todec64x_mul/g
655 s/\<d64xsqrtd128x\>/dec128x_todec64x_sqrt/g
656 s/\<d64xsqrtd128\>/dec128_todec64x_sqrt/g
657 s/\<d64xsqrt\>/todec64x_sqrt/g
658 s/\<d64xsubd128x\>/dec128x_todec64x_sub/g
659 s/\<d64xsubd128\>/dec128_todec64x_sub/g
660 s/\<d64xsub\>/todec64x_sub/g
661 s/\<dadd1\>/ldbl_todbl_add/g
662 s/\<dadd\>/todbl_add/g
663 s/\<ddiv1\>/ldbl_todbl_div/g
664 s/\<ddiv\>/todbl_div/g
665 s/\<decodebind128x\>/dec128x_decodebin/g
666 s/\<decodebind128\>/dec128_decodebin/g
667 s/\<decodebind32\>/dec32_decodebin/g
668 s/\<decodebind64x\>/dec64x_decodebin/g
669 s/\<decodebind64\>/dec64_decodebin/g
670 # no change for decodebin
671 s/\<decodedecd128x\>/dec128x_decodedec/g
672 s/\<decodedecd128\>/dec128_decodedec/g
673 s/\<decodedecd32\>/dec32_decodedec/g
674 s/\<decodedecd64x\>/dec64x_decodedec/g
675 s/\<decodedecd64\>/dec64_decodedec/g
676 # no change for decodedec
677 # no change for deprecated
678 s/\<dfmal\>/ldbl_todbl_fma/g
679 s/\<dfma\>/todbl_fma/g
680 s/\<dmull\>/ldbl_todbl_mul/g
681 s/\<dmul\>/todbl_mul/g
682 s/\<dsqrt1\>/ldbl_todbl_sqrt/g

```

```

683 s\<dsqrt\>/todbl_sqrt/g
684 s\<dsubl\>/ldbl_todbl_sub/g
685 s\<dsub\>/todbl_sub/g
686 s\<encodebind128x\>/dec128x_encodebin/g
687 s\<encodebind128\>/dec128_encodebin/g
688 s\<encodebind32\>/dec32_encodebin/g
689 s\<encodebind64x\>/dec64x_encodebin/g
690 s\<encodebind64\>/dec64_encodebin/g
691 # no change for encodebin
692 s\<encodedecd128x\>/dec128x_encodedec/g
693 s\<encodedecd128\>/dec128_encodedec/g
694 s\<encodedec32\>/dec32_encodedec/g
695 s\<encodedecd64x\>/dec64x_encodedec/g
696 s\<encodedecd64\>/dec64_encodedec/g
697 # no change for encodedec
698 s\<erfcd128x\>/dec128x_erfc/g
699 s\<erfcd128\>/dec128_erfc/g
700 s\<erfcd32\>/dec32_erfc/g
701 s\<erfcd64x\>/dec64x_erfc/g
702 s\<erfcd64\>/dec64_erfc/g
703 s\<erfcf128x\>/flt128x_erfc/g
704 s\<erfcf128\>/flt128_erfc/g
705 s\<erfcf16\>/flt16_erfc/g
706 s\<erfcf32x\>/flt32x_erfc/g
707 s\<erfcf32\>/flt32_erfc/g
708 s\<erfcf64x\>/flt64x_erfc/g
709 s\<erfcf64\>/flt64_erfc/g
710 s\<erfd128x\>/dec128x_erf/g
711 s\<erfd128\>/dec128_erf/g
712 s\<erfd32\>/dec32_erf/g
713 s\<erfd64x\>/dec64x_erf/g
714 s\<erfd64\>/dec64_erf/g
715 s\<erff128x\>/flt128x_erf/g
716 s\<erff128\>/flt128_erf/g
717 s\<erff16\>/flt16_erf/g
718 s\<erff32x\>/flt32x_erf/g
719 s\<erff32\>/flt32_erf/g
720 s\<erff64x\>/flt64x_erf/g
721 s\<erff64\>/flt64_erf/g
722 s\<exp10d128x\>/dec128x_exp10/g
723 s\<exp10d128\>/dec128_exp10/g
724 s\<exp10d32\>/dec32_exp10/g
725 s\<exp10d64x\>/dec64x_exp10/g
726 s\<exp10d64\>/dec64_exp10/g
727 s\<exp10f128x\>/flt128x_exp10/g
728 s\<exp10f128\>/flt128_exp10/g
729 s\<exp10f32x\>/flt32x_exp10/g
730 s\<exp10f32\>/flt32_exp10/g
731 s\<exp10f64x\>/flt64x_exp10/g
732 s\<exp10f64\>/flt64_exp10/g
733 s\<exp10f\>/flt_exp10/g
734 s\<exp10l\>/ldbl_exp10/g
735 s\<exp10md128x\>/dec128x_exp10m1/g
736 s\<exp10md128\>/dec128_exp10m1/g
737 s\<exp10md32\>/dec32_exp10m1/g
738 s\<exp10md64x\>/dec64x_exp10m1/g
739 s\<exp10md64\>/dec64_exp10m1/g
740 s\<exp10mf128x\>/flt128x_exp10m1/g
741 s\<exp10mf128\>/flt128_exp10m1/g
742 s\<exp10mf32x\>/flt32x_exp10m1/g
743 s\<exp10mf32\>/flt32_exp10m1/g
744 s\<exp10mf64x\>/flt64x_exp10m1/g
745 s\<exp10mf64\>/flt64_exp10m1/g

```

```

746 s/\<exp10m1f\>/flt_exp10m1/g
747 s/\<exp10m1l\>/ldbl_exp10m1/g
748 # no change for exp10m1
749 # no change for exp10
750 s/\<exp2d128x\>/dec128x_exp2/g
751 s/\<exp2d128\>/dec128_exp2/g
752 s/\<exp2d32\>/dec32_exp2/g
753 s/\<exp2d64x\>/dec64x_exp2/g
754 s/\<exp2d64\>/dec64_exp2/g
755 s/\<exp2f128x\>/flt128x_exp2/g
756 s/\<exp2f128\>/flt128_exp2/g
757 s/\<exp2f32x\>/flt32x_exp2/g
758 s/\<exp2f32\>/flt32_exp2/g
759 s/\<exp2f64x\>/flt64x_exp2/g
760 s/\<exp2f64\>/flt64_exp2/g
761 s/\<exp2m1d128x\>/dec128x_exp2m1/g
762 s/\<exp2m1d128\>/dec128_exp2m1/g
763 s/\<exp2m1d32\>/dec32_exp2m1/g
764 s/\<exp2m1d64x\>/dec64x_exp2m1/g
765 s/\<exp2m1d64\>/dec64_exp2m1/g
766 s/\<exp2m1f128x\>/flt128x_exp2m1/g
767 s/\<exp2m1f128\>/flt128_exp2m1/g
768 s/\<exp2m1f32x\>/flt32x_exp2m1/g
769 s/\<exp2m1f32\>/flt32_exp2m1/g
770 s/\<exp2m1f64x\>/flt64x_exp2m1/g
771 s/\<exp2m1f64\>/flt64_exp2m1/g
772 s/\<exp2m1f\>/flt_exp2m1/g
773 s/\<exp2m1l\>/ldbl_exp2m1/g
774 # no change for exp2m1
775 s/\<expd128x\>/dec128x_exp/g
776 s/\<expd128\>/dec128_exp/g
777 s/\<expd32\>/dec32_exp/g
778 s/\<expd64x\>/dec64x_exp/g
779 s/\<expd64\>/dec64_exp/g
780 s/\<expf128x\>/flt128x_exp/g
781 s/\<expf128\>/flt128_exp/g
782 s/\<expf16\>/flt16_exp/g
783 s/\<expf32x\>/flt32x_exp/g
784 s/\<expf32\>/flt32_exp/g
785 s/\<expf64x\>/flt64x_exp/g
786 s/\<expf64\>/flt64_exp/g
787 s/\<expm1d128x\>/dec128x_expm1/g
788 s/\<expm1d128\>/dec128_expm1/g
789 s/\<expm1d32\>/dec32_expm1/g
790 s/\<expm1d64x\>/dec64x_expm1/g
791 s/\<expm1d64\>/dec64_expm1/g
792 s/\<expm1f128x\>/flt128x_expm1/g
793 s/\<expm1f128\>/flt128_expm1/g
794 s/\<expm1f32x\>/flt32x_expm1/g
795 s/\<expm1f32\>/flt32_expm1/g
796 s/\<expm1f64x\>/flt64x_expm1/g
797 s/\<expm1f64\>/flt64_expm1/g
798 s/\<f128addf128x\>/flt128x_toflt128_add/g
799 s/\<f128add\>/toflt128_add/g
800 s/\<f128divf128x\>/flt128x_toflt128_div/g
801 s/\<f128div\>/toflt128_div/g
802 s/\<f128maf128x\>/flt128x_toflt128_fma/g
803 s/\<f128fma\>/toflt128_fma/g
804 s/\<f128mulf128x\>/flt128x_toflt128_mul/g
805 s/\<f128mul\>/toflt128_mul/g
806 s/\<f128sqrtf128x\>/flt128x_toflt128_sqrt/g
807 s/\<f128sqrt\>/toflt128_sqrt/g
808 s/\<f128subf128x\>/flt128x_toflt128_sub/g

```

```
809 s\<f128sub\>/toflt128_sub/g
810 s\<f16addf128x\>/flt128x_toflt16_add/g
811 s\<f16addf128\>/flt128_toflt16_add/g
812 s\<f16addf32x\>/flt32x_toflt16_add/g
813 s\<f16addf32\>/flt32_toflt16_add/g
814 s\<f16addf64x\>/flt64x_toflt16_add/g
815 s\<f16addf64\>/flt64_toflt16_add/g
816 s\<f16add\>/toflt16_add/g
817 s\<f16divf128x\>/flt128x_toflt16_div/g
818 s\<f16divf128\>/flt128_toflt16_div/g
819 s\<f16divf32x\>/flt32x_toflt16_div/g
820 s\<f16divf32\>/flt32_toflt16_div/g
821 s\<f16divf64x\>/flt64x_toflt16_div/g
822 s\<f16divf64\>/flt64_toflt16_div/g
823 s\<f16div\>/toflt16_div/g
824 s\<f16maf128x\>/flt128x_toflt16_fma/g
825 s\<f16maf128\>/flt128_toflt16_fma/g
826 s\<f16maf32x\>/flt32x_toflt16_fma/g
827 s\<f16maf32\>/flt32_toflt16_fma/g
828 s\<f16maf64x\>/flt64x_toflt16_fma/g
829 s\<f16maf64\>/flt64_toflt16_fma/g
830 s\<f16fma\>/toflt16_fma/g
831 s\<f16mulf128x\>/flt128x_toflt16_mul/g
832 s\<f16mulf128\>/flt128_toflt16_mul/g
833 s\<f16mulf32x\>/flt32x_toflt16_mul/g
834 s\<f16mulf32\>/flt32_toflt16_mul/g
835 s\<f16mulf64x\>/flt64x_toflt16_mul/g
836 s\<f16mulf64\>/flt64_toflt16_mul/g
837 s\<f16mul\>/toflt16_mul/g
838 s\<f16sqrtf128x\>/flt128x_toflt16_sqrt/g
839 s\<f16sqrtf128\>/flt128_toflt16_sqrt/g
840 s\<f16sqrtf32x\>/flt32x_toflt16_sqrt/g
841 s\<f16sqrtf32\>/flt32_toflt16_sqrt/g
842 s\<f16sqrtf64x\>/flt64x_toflt16_sqrt/g
843 s\<f16sqrtf64\>/flt64_toflt16_sqrt/g
844 s\<f16sqrt\>/toflt16_sqrt/g
845 s\<f16subf128x\>/flt128x_toflt16_sub/g
846 s\<f16subf128\>/flt128_toflt16_sub/g
847 s\<f16subf32x\>/flt32x_toflt16_sub/g
848 s\<f16subf32\>/flt32_toflt16_sub/g
849 s\<f16subf64x\>/flt64x_toflt16_sub/g
850 s\<f16subf64\>/flt64_toflt16_sub/g
851 s\<f16sub\>/toflt16_sub/g
852 s\<f32addf128x\>/flt128x_toflt32_add/g
853 s\<f32addf128\>/flt128_toflt32_add/g
854 s\<f32addf32x\>/flt32x_toflt32_add/g
855 s\<f32addf64x\>/flt64x_toflt32_add/g
856 s\<f32addf64\>/flt64_toflt32_add/g
857 s\<f32add\>/toflt32_add/g
858 s\<f32divf128x\>/flt128x_toflt32_div/g
859 s\<f32divf128\>/flt128_toflt32_div/g
860 s\<f32divf32x\>/flt32x_toflt32_div/g
861 s\<f32divf64x\>/flt64x_toflt32_div/g
862 s\<f32divf64\>/flt64_toflt32_div/g
863 s\<f32div\>/toflt32_div/g
864 s\<f32maf128x\>/flt128x_toflt32_fma/g
865 s\<f32maf128\>/flt128_toflt32_fma/g
866 s\<f32maf32x\>/flt32x_toflt32_fma/g
867 s\<f32maf64x\>/flt64x_toflt32_fma/g
868 s\<f32maf64\>/flt64_toflt32_fma/g
869 s\<f32fma\>/toflt32_fma/g
870 s\<f32mulf128x\>/flt128x_toflt32_mul/g
871 s\<f32mulf128\>/flt128_toflt32_mul/g
```

```
872 s/\<f32mulf32x\>/flt32x_toflt32_mul/g
873 s/\<f32mulf64x\>/flt64x_toflt32_mul/g
874 s/\<f32mulf64\>/flt64_toflt32_mul/g
875 s/\<f32mul\>/toflt32_mul/g
876 s/\<f32sqrtf128x\>/flt128x_toflt32_sqrt/g
877 s/\<f32sqrtf128\>/flt128_toflt32_sqrt/g
878 s/\<f32sqrtf32x\>/flt32x_toflt32_sqrt/g
879 s/\<f32sqrtf64x\>/flt64x_toflt32_sqrt/g
880 s/\<f32sqrtf64\>/flt64_toflt32_sqrt/g
881 s/\<f32sqrt\>/toflt32_sqrt/g
882 s/\<f32subf128x\>/flt128x_toflt32_sub/g
883 s/\<f32subf128\>/flt128_toflt32_sub/g
884 s/\<f32subf32x\>/flt32x_toflt32_sub/g
885 s/\<f32subf64x\>/flt64x_toflt32_sub/g
886 s/\<f32subf64\>/flt64_toflt32_sub/g
887 s/\<f32sub\>/toflt32_sub/g
888 s/\<f32xaddf128x\>/flt128x_toflt32x_add/g
889 s/\<f32xaddf128\>/flt128_toflt32x_add/g
890 s/\<f32xaddf64x\>/flt64x_toflt32x_add/g
891 s/\<f32xaddf64\>/flt64_toflt32x_add/g
892 s/\<f32xadd\>/toflt32x_add/g
893 s/\<f32xdivf128x\>/flt128x_toflt32x_div/g
894 s/\<f32xdivf128\>/flt128_toflt32x_div/g
895 s/\<f32xdivf64x\>/flt64x_toflt32x_div/g
896 s/\<f32xdivf64\>/flt64_toflt32x_div/g
897 s/\<f32xdiv\>/toflt32x_div/g
898 s/\<f32xfmaf128x\>/flt128x_toflt32x_fma/g
899 s/\<f32xfmaf128\>/flt128_toflt32x_fma/g
900 s/\<f32xfmaf64x\>/flt64x_toflt32x_fma/g
901 s/\<f32xfmaf64\>/flt64_toflt32x_fma/g
902 s/\<f32xfma\>/toflt32x_fma/g
903 s/\<f32xmulf128x\>/flt128x_toflt32x_mul/g
904 s/\<f32xmulf128\>/flt128_toflt32x_mul/g
905 s/\<f32xmulf64x\>/flt64x_toflt32x_mul/g
906 s/\<f32xmulf64\>/flt64_toflt32x_mul/g
907 s/\<f32xmul\>/toflt32x_mul/g
908 s/\<f32xsqrtf128x\>/flt128x_toflt32x_sqrt/g
909 s/\<f32xsqrtf128\>/flt128_toflt32x_sqrt/g
910 s/\<f32xsqrtf64x\>/flt64x_toflt32x_sqrt/g
911 s/\<f32xsqrtf64\>/flt64_toflt32x_sqrt/g
912 s/\<f32xsqrt\>/toflt32x_sqrt/g
913 s/\<f32xsubf128x\>/flt128x_toflt32x_sub/g
914 s/\<f32xsubf128\>/flt128_toflt32x_sub/g
915 s/\<f32xsubf64x\>/flt64x_toflt32x_sub/g
916 s/\<f32xsubf64\>/flt64_toflt32x_sub/g
917 s/\<f32xsub\>/toflt32x_sub/g
918 s/\<f64addf128x\>/flt128x_toflt64_add/g
919 s/\<f64addf64x\>/flt64x_toflt64_add/g
920 s/\<f64add\>/toflt64_add/g
921 s/\<f64divf128x\>/flt128x_toflt64_div/g
922 s/\<f64divf64x\>/flt64x_toflt64_div/g
923 s/\<f64div\>/toflt64_div/g
924 s/\<f64fmaf128x\>/flt128x_toflt64_fma/g
925 s/\<f64fmaf64x\>/flt64x_toflt64_fma/g
926 s/\<f64fma\>/toflt64_fma/g
927 s/\<f64mulf128x\>/flt128x_toflt64_mul/g
928 s/\<f64mulf64x\>/flt64x_toflt64_mul/g
929 s/\<f64mul\>/toflt64_mul/g
930 s/\<f64sqrtf128x\>/flt128x_toflt64_sqrt/g
931 s/\<f64sqrtf64x\>/flt64x_toflt64_sqrt/g
932 s/\<f64sqrt\>/toflt64_sqrt/g
933 s/\<f64subf128x\>/flt128x_toflt64_sub/g
934 s/\<f64subf64x\>/flt64x_toflt64_sub/g
```

```

935 s\<f64sub\>/toflt64_sub/g
936 s\<f64xaddf128x\>/flt128x_toflt64x_add/g
937 s\<f64xadd\>/toflt64x_add/g
938 s\<f64xdivf128x\>/flt128x_toflt64x_div/g
939 s\<f64xdiv\>/toflt64x_div/g
940 s\<f64xfmaf128x\>/flt128x_toflt64x_fma/g
941 s\<f64xfma\>/toflt64x_fma/g
942 s\<f64xmulf128x\>/flt128x_toflt64x_mul/g
943 s\<f64xmul\>/toflt64x_mul/g
944 s\<f64xsqrtf128x\>/flt128x_toflt64x_sqrt/g
945 s\<f64xsqrt\>/toflt64x_sqrt/g
946 s\<f64xsubf128x\>/flt128x_toflt64x_sub/g
947 s\<f64xsub\>/toflt64x_sub/g
948 s\<fabsd128x\>/dec128x_fabs/g
949 s\<fabsd128\>/dec128_fabs/g
950 s\<fabsd32\>/dec32_fabs/g
951 s\<fabsd64x\>/dec64x_fabs/g
952 s\<fabsd64\>/dec64_fabs/g
953 s\<fabsf128x\>/flt128x_fabs/g
954 s\<fabsf128\>/flt128_fabs/g
955 s\<fabsf16\>/flt16_fabs/g
956 s\<fabsf32x\>/flt32x_fabs/g
957 s\<fabsf32\>/flt32_fabs/g
958 s\<fabsf64x\>/flt64x_fabs/g
959 s\<fabsf64\>/flt64_fabs/g
960 s\<fadd1\>/ldbl_toflt_add/g
961 s\<fadd\>/toflt_add/g
962 s\<fdimd128x\>/dec128x_fdim/g
963 s\<fdimd128\>/dec128_fdim/g
964 s\<fdimd32\>/dec32_fdim/g
965 s\<fdimd64x\>/dec64x_fdim/g
966 s\<fdimd64\>/dec64_fdim/g
967 s\<fdimf128x\>/flt128x_fdim/g
968 s\<fdimf128\>/flt128_fdim/g
969 s\<fdimf16\>/flt16_fdim/g
970 s\<fdimf32x\>/flt32x_fdim/g
971 s\<fdimf32\>/flt32_fdim/g
972 s\<fdimf64x\>/flt64x_fdim/g
973 s\<fdimf64\>/flt64_fdim/g
974 s\<fdiv1\>/ldbl_toflt_div/g
975 s\<fdiv\>/toflt_div/g
976 # no change for fe_dec_getround
977 # no change for fe_dec_setround
978 s\<fegetmode\>/fe_getmode/g
979 # no change for femode_t
980 s\<fesetexcept\>/fe_setexcept/g
981 s\<fesetmode\>/fe_setmode/g
982 s\<fetetestexceptflag\>/fe_testexceptflag/g
983 s\<ffmal\>/ldbl_toflt_fma/g
984 s\<ffma\>/toflt_fma/g
985 s\<floord128x\>/dec128x_floor/g
986 s\<floord128\>/dec128_floor/g
987 s\<floord32\>/dec32_floor/g
988 s\<floorf128x\>/flt128x_floor/g
989 s\<floorf16\>/flt16_floor/g
990 s\<floorf32x\>/flt32x_floor/g
991 s\<floorf32\>/flt32_floor/g
992 s\<fmad128x\>/dec128x_fma/g
993 s\<fmad128\>/dec128_fma/g
994 s\<fmad32\>/dec32_fma/g
995 s\<fmad64x\>/dec64x_fma/g
996 s\<fmad64\>/dec64_fma/g
997 s\<fmalf128x\>/flt128x_fma/g

```



```

998 s/\<fmaf128\>/flt128_fma/g
999 s/\<fmaf16\>/flt16_fma/g
1000 s/\<fmaf32x\>/flt32x_fma/g
1001 s/\<fmaf32\>/flt32_fma/g
1002 s/\<fmaf64x\>/flt64x_fma/g
1003 s/\<fmaf64\>/flt64_fma/g
1004 s/\<fmaxd128x\>/dec128x_fmax/g
1005 s/\<fmaxd128\>/dec128_fmax/g
1006 s/\<fmaxd32\>/dec32_fmax/g
1007 s/\<fmaxd64x\>/dec64x_fmax/g
1008 s/\<fmaxd64\>/dec64_fmax/g
1009 s/\<fmaxf128x\>/flt128x_fmax/g
1010 s/\<fmaxf128\>/flt128_fmax/g
1011 s/\<fmaxf16\>/flt16_fmax/g
1012 s/\<fmaxf32x\>/flt32x_fmax/g
1013 s/\<fmaxf32\>/flt32_fmax/g
1014 s/\<fmaxf64x\>/flt64x_fmax/g
1015 s/\<fmaxf64\>/flt64_fmax/g
1016 s/\<fmaxmagd128x\>/dec128x_fmaxmag/g
1017 s/\<fmaxmagd128\>/dec128_fmaxmag/g
1018 s/\<fmaxmagd32\>/dec32_fmaxmag/g
1019 s/\<fmaxmagd64x\>/dec64x_fmaxmag/g
1020 s/\<fmaxmagd64\>/dec64_fmaxmag/g
1021 s/\<fmaxmagf128x\>/flt128x_fmaxmag/g
1022 s/\<fmaxmagf128\>/flt128_fmaxmag/g
1023 s/\<fmaxmagf16\>/flt16_fmaxmag/g
1024 s/\<fmaxmagf32x\>/flt32x_fmaxmag/g
1025 s/\<fmaxmagf32\>/flt32_fmaxmag/g
1026 s/\<fmaxmagf64x\>/flt64x_fmaxmag/g
1027 s/\<fmaxmagf64\>/flt64_fmaxmag/g
1028 s/\<fmaxmagf\>/flt_fmaxmag/g
1029 s/\<fmaxmagl\>/ldbl_fmaxmag/g
1030 # no change for fmaxmag
1031 s/\<fmind128x\>/dec128x_fmin/g
1032 s/\<fmind128\>/dec128_fmin/g
1033 s/\<fmind32\>/dec32_fmin/g
1034 s/\<fmind64x\>/dec64x_fmin/g
1035 s/\<fmind64\>/dec64_fmin/g
1036 s/\<fminf128x\>/flt128x_fmin/g
1037 s/\<fminf128\>/flt128_fmin/g
1038 s/\<fminf16\>/flt16_fmin/g
1039 s/\<fminf32x\>/flt32x_fmin/g
1040 s/\<fminf32\>/flt32_fmin/g
1041 s/\<fminf64x\>/flt64x_fmin/g
1042 s/\<fminf64\>/flt64_fmin/g
1043 s/\<fminmagd128x\>/dec128x_fminmag/g
1044 s/\<fminmagd128\>/dec128_fminmag/g
1045 s/\<fminmagd32\>/dec32_fminmag/g
1046 s/\<fminmagd64x\>/dec64x_fminmag/g
1047 s/\<fminmagd64\>/dec64_fminmag/g
1048 s/\<fminmagf128x\>/flt128x_fminmag/g
1049 s/\<fminmagf128\>/flt128_fminmag/g
1050 s/\<fminmagf16\>/flt16_fminmag/g
1051 s/\<fminmagf32x\>/flt32x_fminmag/g
1052 s/\<fminmagf32\>/flt32_fminmag/g
1053 s/\<fminmagf64x\>/flt64x_fminmag/g
1054 s/\<fminmagf64\>/flt64_fminmag/g
1055 s/\<fminmagf\>/flt_fminmag/g
1056 s/\<fminmagl\>/ldbl_fminmag/g
1057 # no change for fminmag
1058 s/\<fmodd128x\>/dec128x_fmod/g
1059 s/\<fmodd128\>/dec128_fmod/g
1060 s/\<fmodd32\>/dec32_fmod/g

```

```
1061 s\<fmodd64x\>/dec64x_fmod/g
1062 s\<fmodd64\>/dec64_fmod/g
1063 s\<fmodf128x\>/flt128x_fmod/g
1064 s\<fmodf128\>/flt128_fmod/g
1065 s\<fmodf16\>/flt16_fmod/g
1066 s\<fmodf32x\>/flt32x_fmod/g
1067 s\<fmodf32\>/flt32_fmod/g
1068 s\<fmodf64x\>/flt64x_fmod/g
1069 s\<fmodf64\>/flt64_fmod/g
1070 s\<fmull\>/ldbl_toflt_mul/g
1071 s\<fmul\>/toflt_mul/g
1072 s\<frexpd128x\>/dec128x_frexp/g
1073 s\<frexpd128\>/dec128_frexp/g
1074 s\<frexpd32\>/dec32_frexp/g
1075 s\<frexpd64x\>/dec64x_frexp/g
1076 s\<frexpd64\>/dec64_frexp/g
1077 s\<frexpf128x\>/flt128x_frexp/g
1078 s\<frexpf128\>/flt128_frexp/g
1079 s\<frexpf16\>/flt16_frexp/g
1080 s\<frexpf32x\>/flt32x_frexp/g
1081 s\<frexpf32\>/flt32_frexp/g
1082 s\<frexpf64x\>/flt64x_frexp/g
1083 s\<frexpf64\>/flt64_frexp/g
1084 s\<fromfpd128x\>/dec128x_toint/g
1085 s\<fromfpd128\>/dec128_toint/g
1086 s\<fromfpd32\>/dec32_toint/g
1087 s\<fromfpd64x\>/dec64x_toint/g
1088 s\<fromfpd64\>/dec64_toint/g
1089 s\<fromfpf128x\>/flt128x_toint/g
1090 s\<fromfpf128\>/flt128_toint/g
1091 s\<fromfpf16\>/flt16_toint/g
1092 s\<fromfpf32x\>/flt32x_toint/g
1093 s\<fromfpf32\>/flt32_toint/g
1094 s\<fromfpf64x\>/flt64x_toint/g
1095 s\<fromfpf64\>/flt64_toint/g
1096 s\<fromfpf\>/flt_toint/g
1097 s\<fromfpl\>/ldbl_toint/g
1098 s\<fromfpxd128x\>/dec128x_tointx/g
1099 s\<fromfpxd128\>/dec128_tointx/g
1100 s\<fromfpxd32\>/dec32_tointx/g
1101 s\<fromfpxd64x\>/dec64x_tointx/g
1102 s\<fromfpxd64\>/dec64_tointx/g
1103 s\<fromfpxf128x\>/flt128x_tointx/g
1104 s\<fromfpxf128\>/flt128_tointx/g
1105 s\<fromfpxf16\>/flt16_tointx/g
1106 s\<fromfpxf32x\>/flt32x_tointx/g
1107 s\<fromfpxf32\>/flt32_tointx/g
1108 s\<fromfpxf64x\>/flt64x_tointx/g
1109 s\<fromfpxf64\>/flt64_tointx/g
1110 s\<fromfpxf\>/flt_tointx/g
1111 s\<fromfpx1\>/ldbl_tointx/g
1112 s\<fromfpx\>/tointx/g
1113 s\<fromfp\>/toint/g
1114 s\<fsqrt1\>/ldbl_toflt_sqrt/g
1115 s\<fsqrt\>/toflt_sqrt/g
1116 s\<fsub1\>/ldbl_toflt_sub/g
1117 s\<fsub\>/toflt_sub/g
1118 s\<getpayloadd128x\>/dec128x_getpayload/g
1119 s\<getpayloadd128\>/dec128_getpayload/g
1120 s\<getpayloadd32\>/dec32_getpayload/g
1121 s\<getpayloadd64x\>/dec64x_getpayload/g
1122 s\<getpayloadd64\>/dec64_getpayload/g
1123 s\<getpayloadf128x\>/flt128x_getpayload/g
```

```
1124 s/\<getpayloadf128\>/flt128_getpayload/g
1125 s/\<getpayloadf16\>/flt16_getpayload/g
1126 s/\<getpayloadf32x\>/flt32x_getpayload/g
1127 s/\<getpayloadf32\>/flt32_getpayload/g
1128 s/\<getpayloadf64x\>/flt64x_getpayload/g
1129 s/\<getpayloadf64\>/flt64_getpayload/g
1130 s/\<getpayloadf\>/flt_getpayload/g
1131 s/\<getpayloadl\>/ldbl_getpayload/g
1132 # no change for getpayload
1133 s/\<hypotd128x\>/dec128x_hypot/g
1134 s/\<hypotd128\>/dec128_hypot/g
1135 s/\<hypotd32\>/dec32_hypot/g
1136 s/\<hypotd64x\>/dec64x_hypot/g
1137 s/\<hypotd64\>/dec64_hypot/g
1138 s/\<hypotf128x\>/flt128x_hypot/g
1139 s/\<hypotf128\>/flt128_hypot/g
1140 s/\<hypotf16\>/flt16_hypot/g
1141 s/\<hypotf32x\>/flt32x_hypot/g
1142 s/\<hypotf32\>/flt32_hypot/g
1143 s/\<hypotf64x\>/flt64x_hypot/g
1144 s/\<hypotf64\>/flt64_hypot/g
1145 s/\<ilogbd128x\>/dec128x_ilogb/g
1146 s/\<ilogbd128\>/dec128_ilogb/g
1147 s/\<ilogbd32\>/dec32_ilogb/g
1148 s/\<ilogbd64x\>/dec64x_ilogb/g
1149 s/\<ilogbd64\>/dec64_ilogb/g
1150 s/\<ilogbf128x\>/flt128x_ilogb/g
1151 s/\<ilogbf128\>/flt128_ilogb/g
1152 s/\<ilogbf16\>/flt16_ilogb/g
1153 s/\<ilogbf32x\>/flt32x_ilogb/g
1154 s/\<ilogbf32\>/flt32_ilogb/g
1155 s/\<ilogbf64x\>/flt64x_ilogb/g
1156 s/\<ilogbf64\>/flt64_ilogb/g
1157 s/\<ldexpd128x\>/dec128x_ldexp/g
1158 s/\<ldexpd128\>/dec128_ldexp/g
1159 s/\<ldexpd32\>/dec32_ldexp/g
1160 s/\<ldexpd64x\>/dec64x_ldexp/g
1161 s/\<ldexpd64\>/dec64_ldexp/g
1162 s/\<ldexpf128x\>/flt128x_ldexp/g
1163 s/\<ldexpf128\>/flt128_ldexp/g
1164 s/\<ldexpf16\>/flt16_ldexp/g
1165 s/\<ldexpf32x\>/flt32x_ldexp/g
1166 s/\<ldexpf32\>/flt32_ldexp/g
1167 s/\<ldexpf64x\>/flt64x_ldexp/g
1168 s/\<ldexpf64\>/flt64_ldexp/g
1169 s/\<lgammad128x\>/dec128x_lgamma/g
1170 s/\<lgammad128\>/dec128_lgamma/g
1171 s/\<lgammad32\>/dec32_lgamma/g
1172 s/\<lgammad64x\>/dec64x_lgamma/g
1173 s/\<lgammad64\>/dec64_lgamma/g
1174 s/\<lgammaf128x\>/flt128x_lgamma/g
1175 s/\<lgammaf128\>/flt128_lgamma/g
1176 s/\<lgammaf16\>/flt16_lgamma/g
1177 s/\<lgammaf32x\>/flt32x_lgamma/g
1178 s/\<lgammaf32\>/flt32_lgamma/g
1179 s/\<lgammaf64x\>/flt64x_lgamma/g
1180 s/\<lgammaf64\>/flt64_lgamma/g
1181 s/\<llogbd128x\>/dec128x_toint_logb/g
1182 s/\<llogbd128\>/dec128_toint_logb/g
1183 s/\<llogbd32\>/dec32_toint_logb/g
1184 s/\<llogbd64x\>/dec64x_toint_logb/g
1185 s/\<llogbd64\>/dec64_toint_logb/g
1186 s/\<llogbf128x\>/flt128x_toint_logb/g
```

```
1187 s^\<llogbf128\>/flt128_toint_logb/g
1188 s^\<llogbf16\>/flt16_toint_logb/g
1189 s^\<llogbf32x\>/flt32x_toint_logb/g
1190 s^\<llogbf32\>/flt32_toint_logb/g
1191 s^\<llogbf64x\>/flt64x_toint_logb/g
1192 s^\<llogbf64\>/flt64_toint_logb/g
1193 s^\<llogbf\>/flt_toint_logb/g
1194 s^\<llogbl\>/ldbl_toint_logb/g
1195 s^\<llogb\>/toint_logb/g
1196 s^\<llquantexpd128x\>/dec128x_toint_quantexp/g
1197 s^\<llquantexpd128\>/dec128_toint_quantexp/g
1198 s^\<llquantexpd32\>/dec32_toint_quantexp/g
1199 s^\<llquantexpd64x\>/dec64x_toint_quantexp/g
1200 s^\<llquantexpd64\>/dec64_toint_quantexp/g
1201 s^\<llquantexpf128x\>/flt128x_toint_quantexp/g
1202 s^\<llquantexpf128\>/flt128_toint_quantexp/g
1203 s^\<llquantexpf16\>/flt16_toint_quantexp/g
1204 s^\<llquantexpf32x\>/flt32x_toint_quantexp/g
1205 s^\<llquantexpf32\>/flt32_toint_quantexp/g
1206 s^\<llquantexpf64x\>/flt64x_toint_quantexp/g
1207 s^\<llquantexpf64\>/flt64_toint_quantexp/g
1208 s^\<llquantexp\>/toint_quantexp/g
1209 s^\<llrintd128x\>/dec128x_llrint/g
1210 s^\<llrintd128\>/dec128_llrint/g
1211 s^\<llrintd32\>/dec32_llrint/g
1212 s^\<llrintd64x\>/dec64x_llrint/g
1213 s^\<llrintd64\>/dec64_llrint/g
1214 s^\<llrintf128x\>/flt128x_llrint/g
1215 s^\<llrintf128\>/flt128_llrint/g
1216 s^\<llrintf16\>/flt16_llrint/g
1217 s^\<llrintf32x\>/flt32x_llrint/g
1218 s^\<llrintf32\>/flt32_llrint/g
1219 s^\<llrintf64x\>/flt64x_llrint/g
1220 s^\<llrintf64\>/flt64_llrint/g
1221 s^\<llroundd128x\>/dec128x_llround/g
1222 s^\<llroundd128\>/dec128_llround/g
1223 s^\<llroundd32\>/dec32_llround/g
1224 s^\<llroundd64x\>/dec64x_llround/g
1225 s^\<llroundd64\>/dec64_llround/g
1226 s^\<llroundf128x\>/flt128x_llround/g
1227 s^\<llroundf128\>/flt128_llround/g
1228 s^\<llroundf16\>/flt16_llround/g
1229 s^\<llroundf32x\>/flt32x_llround/g
1230 s^\<llroundf32\>/flt32_llround/g
1231 s^\<llroundf64x\>/flt64x_llround/g
1232 s^\<llroundf64\>/flt64_llround/g
1233 s^\<log10d128x\>/dec128x_log10/g
1234 s^\<log10d128\>/dec128_log10/g
1235 s^\<log10d32\>/dec32_log10/g
1236 s^\<log10d64x\>/dec64x_log10/g
1237 s^\<log10d64\>/dec64_log10/g
1238 s^\<log10f128x\>/flt128x_log10/g
1239 s^\<log10f128\>/flt128_log10/g
1240 s^\<log10f32x\>/flt32x_log10/g
1241 s^\<log10f32\>/flt32_log10/g
1242 s^\<log10f64x\>/flt64x_log10/g
1243 s^\<log10f64\>/flt64_log10/g
1244 s^\<log10p1d128x\>/dec128x_log10p1/g
1245 s^\<log10p1d128\>/dec128_log10p1/g
1246 s^\<log10p1d32\>/dec32_log10p1/g
1247 s^\<log10p1d64x\>/dec64x_log10p1/g
1248 s^\<log10p1d64\>/dec64_log10p1/g
1249 s^\<log10p1f128x\>/flt128x_log10p1/g
```

```

1250 s/\<log10p1f128\>/flt128_log10p1/g
1251 s/\<log10p1f32x\>/flt32x_log10p1/g
1252 s/\<log10p1f32\>/flt32_log10p1/g
1253 s/\<log10p1f64x\>/flt64x_log10p1/g
1254 s/\<log10p1f64\>/flt64_log10p1/g
1255 s/\<log10p1f\>/flt_log10p1/g
1256 s/\<log10p1l\>/ldbl_log10p1/g
1257 # no change for log10p1
1258 s/\<log1pd128x\>/dec128x_log1p/g
1259 s/\<log1pd128\>/dec128_log1p/g
1260 s/\<log1pd32\>/dec32_log1p/g
1261 s/\<log1pd64x\>/dec64x_log1p/g
1262 s/\<log1pd64\>/dec64_log1p/g
1263 s/\<log1pf128x\>/flt128x_log1p/g
1264 s/\<log1pf128\>/flt128_log1p/g
1265 s/\<log1pf32x\>/flt32x_log1p/g
1266 s/\<log1pf32\>/flt32_log1p/g
1267 s/\<log1pf64x\>/flt64x_log1p/g
1268 s/\<log1pf64\>/flt64_log1p/g
1269 s/\<log2d128x\>/dec128x_log2/g
1270 s/\<log2d128\>/dec128_log2/g
1271 s/\<log2d32\>/dec32_log2/g
1272 s/\<log2d64x\>/dec64x_log2/g
1273 s/\<log2d64\>/dec64_log2/g
1274 s/\<log2f128x\>/flt128x_log2/g
1275 s/\<log2f128\>/flt128_log2/g
1276 s/\<log2f32x\>/flt32x_log2/g
1277 s/\<log2f32\>/flt32_log2/g
1278 s/\<log2f64x\>/flt64x_log2/g
1279 s/\<log2f64\>/flt64_log2/g
1280 s/\<log2pd128x\>/dec128x_log2p1/g
1281 s/\<log2pd128\>/dec128_log2p1/g
1282 s/\<log2pd32\>/dec32_log2p1/g
1283 s/\<log2pd64x\>/dec64x_log2p1/g
1284 s/\<log2pd64\>/dec64_log2p1/g
1285 s/\<log2p1f128x\>/flt128x_log2p1/g
1286 s/\<log2p1f128\>/flt128_log2p1/g
1287 s/\<log2p1f32x\>/flt32x_log2p1/g
1288 s/\<log2p1f32\>/flt32_log2p1/g
1289 s/\<log2p1f64x\>/flt64x_log2p1/g
1290 s/\<log2p1f64\>/flt64_log2p1/g
1291 s/\<log2p1f\>/flt_log2p1/g
1292 s/\<log2p1l\>/ldbl_log2p1/g
1293 # no change for log2p1
1294 s/\<logbd128x\>/dec128x_logb/g
1295 s/\<logbd128\>/dec128_logb/g
1296 s/\<logbd32\>/dec32_logb/g
1297 s/\<logbd64x\>/dec64x_logb/g
1298 s/\<logbd64\>/dec64_logb/g
1299 s/\<logbf128x\>/flt128x_logb/g
1300 s/\<logbf128\>/flt128_logb/g
1301 s/\<logbf16\>/flt16_logb/g
1302 s/\<logbf32x\>/flt32x_logb/g
1303 s/\<logbf32\>/flt32_logb/g
1304 s/\<logbf64x\>/flt64x_logb/g
1305 s/\<logbf64\>/flt64_logb/g
1306 s/\<logd128x\>/dec128x_log/g
1307 s/\<logd128\>/dec128_log/g
1308 s/\<logd32\>/dec32_log/g
1309 s/\<logd64x\>/dec64x_log/g
1310 s/\<logd64\>/dec64_log/g
1311 s/\<logf128x\>/flt128x_log/g
1312 s/\<logf128\>/flt128_log/g

```

```

1313 s^\<logf16\>/flt16_log/g
1314 s^\<logf32x\>/flt32x_log/g
1315 s^\<logf32\>/flt32_log/g
1316 s^\<logf64x\>/flt64x_log/g
1317 s^\<logf64\>/flt64_log/g
1318 s^\<logp1d128x\>/dec128x_logp1/g
1319 s^\<logp1d128\>/dec128_logp1/g
1320 s^\<logp1d32\>/dec32_logp1/g
1321 s^\<logp1d64x\>/dec64x_logp1/g
1322 s^\<logp1d64\>/dec64_logp1/g
1323 s^\<logp1f128x\>/flt128x_logp1/g
1324 s^\<logp1f128\>/flt128_logp1/g
1325 s^\<logp1f32x\>/flt32x_logp1/g
1326 s^\<logp1f32\>/flt32_logp1/g
1327 s^\<logp1f64x\>/flt64x_logp1/g
1328 s^\<logp1f64\>/flt64_logp1/g
1329 s^\<logp1f\>/flt_logp1/g
1330 s^\<logp1l\>/ldbl_logp1/g
1331 # no change for logp1
1332 # no change for long_double_t
1333 s^\<lrintd128x\>/dec128x_lrint/g
1334 s^\<lrintd128\>/dec128_lrint/g
1335 s^\<lrintd32\>/dec32_lrint/g
1336 s^\<lrintd64x\>/dec64x_lrint/g
1337 s^\<lrintd64\>/dec64_lrint/g
1338 s^\<lrintf128x\>/flt128x_lrint/g
1339 s^\<lrintf128\>/flt128_lrint/g
1340 s^\<lrintf16\>/flt16_lrint/g
1341 s^\<lrintf32x\>/flt32x_lrint/g
1342 s^\<lrintf32\>/flt32_lrint/g
1343 s^\<lrintf64x\>/flt64x_lrint/g
1344 s^\<lrintf64\>/flt64_lrint/g
1345 s^\<lroundd128x\>/dec128x_lround/g
1346 s^\<lroundd128\>/dec128_lround/g
1347 s^\<lroundd32\>/dec32_lround/g
1348 s^\<lroundd64x\>/dec64x_lround/g
1349 s^\<lroundd64\>/dec64_lround/g
1350 s^\<lroundf128x\>/flt128x_lround/g
1351 s^\<lroundf128\>/flt128_lround/g
1352 s^\<lroundf16\>/flt16_lround/g
1353 s^\<lroundf32x\>/flt32x_lround/g
1354 s^\<lroundf32\>/flt32_lround/g
1355 s^\<lroundf64x\>/flt64x_lround/g
1356 s^\<lroundf64\>/flt64_lround/g
1357 # no change for maybe_unused
1358 s^\<modfd128x\>/dec128x_modf/g
1359 s^\<modfd128\>/dec128_modf/g
1360 s^\<modfd32\>/dec32_modf/g
1361 s^\<modfd64x\>/dec64x_modf/g
1362 s^\<modfd64\>/dec64_modf/g
1363 s^\<modff128x\>/flt128x_modf/g
1364 s^\<modff128\>/flt128_modf/g
1365 s^\<modff16\>/flt16_modf/g
1366 s^\<modff32x\>/flt32x_modf/g
1367 s^\<modff32\>/flt32_modf/g
1368 s^\<modff64x\>/flt64x_modf/g
1369 s^\<modff64\>/flt64_modf/g
1370 s^\<nand128x\>/dec128x_nan/g
1371 s^\<nand128\>/dec128_nan/g
1372 s^\<nand32\>/dec32_nan/g
1373 s^\<nand64x\>/dec64x_nan/g
1374 s^\<nand64\>/dec64_nan/g
1375 s^\<nanf128x\>/flt128x_nan/g

```

```
1376 s/\<nanf128\>/flt128_nan/g
1377 s/\<nanf16\>/flt16_nan/g
1378 s/\<nanf32x\>/flt32x_nan/g
1379 s/\<nanf32\>/flt32_nan/g
1380 s/\<nanf64x\>/flt64x_nan/g
1381 s/\<nanf64\>/flt64_nan/g
1382 s/\<nearbyintd128x\>/dec128x_nearbyint/g
1383 s/\<nearbyintd128\>/dec128_nearbyint/g
1384 s/\<nearbyintd32\>/dec32_nearbyint/g
1385 s/\<nearbyintd64x\>/dec64x_nearbyint/g
1386 s/\<nearbyintd64\>/dec64_nearbyint/g
1387 s/\<nearbyintf128x\>/flt128x_nearbyint/g
1388 s/\<nearbyintf128\>/flt128_nearbyint/g
1389 s/\<nearbyintf16\>/flt16_nearbyint/g
1390 s/\<nearbyintf32x\>/flt32x_nearbyint/g
1391 s/\<nearbyintf32\>/flt32_nearbyint/g
1392 s/\<nearbyintf64x\>/flt64x_nearbyint/g
1393 s/\<nearbyintf64\>/flt64_nearbyint/g
1394 s/\<nextafterd128x\>/dec128x_nextafter/g
1395 s/\<nextafterd128\>/dec128_nextafter/g
1396 s/\<nextafterd32\>/dec32_nextafter/g
1397 s/\<nextafterd64x\>/dec64x_nextafter/g
1398 s/\<nextafterd64\>/dec64_nextafter/g
1399 s/\<nextafterf128x\>/flt128x_nextafter/g
1400 s/\<nextafterf128\>/flt128_nextafter/g
1401 s/\<nextafterf16\>/flt16_nextafter/g
1402 s/\<nextafterf32x\>/flt32x_nextafter/g
1403 s/\<nextafterf32\>/flt32_nextafter/g
1404 s/\<nextafterf64x\>/flt64x_nextafter/g
1405 s/\<nextafterf64\>/flt64_nextafter/g
1406 s/\<nextdownd128x\>/dec128x_nextdown/g
1407 s/\<nextdownd128\>/dec128_nextdown/g
1408 s/\<nextdownd32\>/dec32_nextdown/g
1409 s/\<nextdownd64x\>/dec64x_nextdown/g
1410 s/\<nextdownd64\>/dec64_nextdown/g
1411 s/\<nextdownf128x\>/flt128x_nextdown/g
1412 s/\<nextdownf128\>/flt128_nextdown/g
1413 s/\<nextdownf16\>/flt16_nextdown/g
1414 s/\<nextdownf32x\>/flt32x_nextdown/g
1415 s/\<nextdownf32\>/flt32_nextdown/g
1416 s/\<nextdownf64x\>/flt64x_nextdown/g
1417 s/\<nextdownf64\>/flt64_nextdown/g
1418 s/\<nextdownf\>/flt_nextdown/g
1419 s/\<nextdown1\>/ldbl_nextdown/g
1420 # no change for nextdown
1421 s/\<nexttowardd128x\>/dec128x_nexttoward/g
1422 s/\<nexttowardd128\>/dec128_nexttoward/g
1423 s/\<nexttowardd32\>/dec32_nexttoward/g
1424 s/\<nexttowardd64x\>/dec64x_nexttoward/g
1425 s/\<nexttowardd64\>/dec64_nexttoward/g
1426 s/\<nexttowardf128x\>/flt128x_nexttoward/g
1427 s/\<nexttowardf128\>/flt128_nexttoward/g
1428 s/\<nexttowardf16\>/flt16_nexttoward/g
1429 s/\<nexttowardf32x\>/flt32x_nexttoward/g
1430 s/\<nexttowardf32\>/flt32_nexttoward/g
1431 s/\<nexttowardf64x\>/flt64x_nexttoward/g
1432 s/\<nexttowardf64\>/flt64_nexttoward/g
1433 s/\<nextupd128x\>/dec128x_nextup/g
1434 s/\<nextupd128\>/dec128_nextup/g
1435 s/\<nextupd32\>/dec32_nextup/g
1436 s/\<nextupd64x\>/dec64x_nextup/g
1437 s/\<nextupd64\>/dec64_nextup/g
1438 s/\<nextupf128x\>/flt128x_nextup/g
```

```
1439 s\<nextupf128\>/flt128_nextup/g
1440 s\<nextupf16\>/flt16_nextup/g
1441 s\<nextupf32x\>/flt32x_nextup/g
1442 s\<nextupf32\>/flt32_nextup/g
1443 s\<nextupf64x\>/flt64x_nextup/g
1444 s\<nextupf64\>/flt64_nextup/g
1445 s\<nextupf\>/flt_nextup/g
1446 s\<nextupl\>/ldbl_nextup/g
1447 # no change for nextup
1448 # no change for nodiscard
1449 s\<powd128x\>/dec128x_pow/g
1450 s\<powd128\>/dec128_pow/g
1451 s\<powd32\>/dec32_pow/g
1452 s\<powd64x\>/dec64x_pow/g
1453 s\<powd64\>/dec64_pow/g
1454 s\<powf128x\>/flt128x_pow/g
1455 s\<powf128\>/flt128_pow/g
1456 s\<powf16\>/flt16_pow/g
1457 s\<powf32x\>/flt32x_pow/g
1458 s\<powf32\>/flt32_pow/g
1459 s\<powf64x\>/flt64x_pow/g
1460 s\<powf64\>/flt64_pow/g
1461 s\<pownd128x\>/dec128x_pown/g
1462 s\<pownd128\>/dec128_pown/g
1463 s\<pownd32\>/dec32_pown/g
1464 s\<pownd64x\>/dec64x_pown/g
1465 s\<pownd64\>/dec64_pown/g
1466 s\<pownf128x\>/flt128x_pown/g
1467 s\<pownf128\>/flt128_pown/g
1468 s\<pownf16\>/flt16_pown/g
1469 s\<pownf32x\>/flt32x_pown/g
1470 s\<pownf32\>/flt32_pown/g
1471 s\<pownf64x\>/flt64x_pown/g
1472 s\<pownf64\>/flt64_pown/g
1473 s\<pownf\>/flt_pown/g
1474 s\<pownl\>/ldbl_pown/g
1475 # no change for pown
1476 s\<powrd128x\>/dec128x_powr/g
1477 s\<powrd128\>/dec128_powr/g
1478 s\<powrd32\>/dec32_powr/g
1479 s\<powrd64x\>/dec64x_powr/g
1480 s\<powrd64\>/dec64_powr/g
1481 s\<powrf128x\>/flt128x_powr/g
1482 s\<powrf128\>/flt128_powr/g
1483 s\<powrf16\>/flt16_powr/g
1484 s\<powrf32x\>/flt32x_powr/g
1485 s\<powrf32\>/flt32_powr/g
1486 s\<powrf64x\>/flt64x_powr/g
1487 s\<powrf64\>/flt64_powr/g
1488 s\<powrf\>/flt_powr/g
1489 s\<powrl\>/ldbl_powr/g
1490 # no change for powr
1491 s\<quantized128x\>/dec128x_quantize/g
1492 s\<quantized128\>/dec128_quantize/g
1493 s\<quantized32\>/dec32_quantize/g
1494 s\<quantized64x\>/dec64x_quantize/g
1495 s\<quantized64\>/dec64_quantize/g
1496 s\<quantizef128x\>/flt128x_quantize/g
1497 s\<quantizef128\>/flt128_quantize/g
1498 s\<quantizef16\>/flt16_quantize/g
1499 s\<quantizef32x\>/flt32x_quantize/g
1500 s\<quantizef32\>/flt32_quantize/g
1501 s\<quantizef64x\>/flt64x_quantize/g
```



```

1502 s/\<quantizef64\>/flt64_quantize/g
1503 # no change for quantize
1504 s/\<quantumd128x\>/dec128x_quantum/g
1505 s/\<quantumd128\>/dec128_quantum/g
1506 s/\<quantumd32\>/dec32_quantum/g
1507 s/\<quantumd64x\>/dec64x_quantum/g
1508 s/\<quantumd64\>/dec64_quantum/g
1509 s/\<quantumf128x\>/flt128x_quantum/g
1510 s/\<quantumf128\>/flt128_quantum/g
1511 s/\<quantumf16\>/flt16_quantum/g
1512 s/\<quantumf32x\>/flt32x_quantum/g
1513 s/\<quantumf32\>/flt32_quantum/g
1514 s/\<quantumf64x\>/flt64x_quantum/g
1515 s/\<quantumf64\>/flt64_quantum/g
1516 # no change for quantum
1517 s/\<remainderd128x\>/dec128x_remainder/g
1518 s/\<remainderd128\>/dec128_remainder/g
1519 s/\<remainderd32\>/dec32_remainder/g
1520 s/\<remainderd64x\>/dec64x_remainder/g
1521 s/\<remainderd64\>/dec64_remainder/g
1522 s/\<remainderf128x\>/flt128x_remainder/g
1523 s/\<remainderf128\>/flt128_remainder/g
1524 s/\<remainderf16\>/flt16_remainder/g
1525 s/\<remainderf32x\>/flt32x_remainder/g
1526 s/\<remainderf32\>/flt32_remainder/g
1527 s/\<remainderf64x\>/flt64x_remainder/g
1528 s/\<remainderf64\>/flt64_remainder/g
1529 s/\<rintd128x\>/dec128x_rint/g
1530 s/\<rintd128\>/dec128_rint/g
1531 s/\<rintd32\>/dec32_rint/g
1532 s/\<rintd64x\>/dec64x_rint/g
1533 s/\<rintd64\>/dec64_rint/g
1534 s/\<rintf128x\>/flt128x_rint/g
1535 s/\<rintf128\>/flt128_rint/g
1536 s/\<rintf16\>/flt16_rint/g
1537 s/\<rintf32x\>/flt32x_rint/g
1538 s/\<rintf32\>/flt32_rint/g
1539 s/\<rintf64x\>/flt64x_rint/g
1540 s/\<rintf64\>/flt64_rint/g
1541 s/\<rootnd128x\>/dec128x_rootn/g
1542 s/\<rootnd128\>/dec128_rootn/g
1543 s/\<rootnd32\>/dec32_rootn/g
1544 s/\<rootnd64x\>/dec64x_rootn/g
1545 s/\<rootnd64\>/dec64_rootn/g
1546 s/\<rootnf128x\>/flt128x_rootn/g
1547 s/\<rootnf128\>/flt128_rootn/g
1548 s/\<rootnf16\>/flt16_rootn/g
1549 s/\<rootnf32x\>/flt32x_rootn/g
1550 s/\<rootnf32\>/flt32_rootn/g
1551 s/\<rootnf64x\>/flt64x_rootn/g
1552 s/\<rootnf64\>/flt64_rootn/g
1553 s/\<rootnf\>/flt_rootn/g
1554 s/\<rootnl\>/ldbl_rootn/g
1555 # no change for rootn
1556 s/\<roundd128x\>/dec128x_round/g
1557 s/\<roundd128\>/dec128_round/g
1558 s/\<roundd32\>/dec32_round/g
1559 s/\<roundd64x\>/dec64x_round/g
1560 s/\<roundd64\>/dec64_round/g
1561 s/\<roundevend128x\>/dec128x_roundeven/g
1562 s/\<roundevend128\>/dec128_roundeven/g
1563 s/\<roundevend32\>/dec32_roundeven/g
1564 s/\<roundevend64x\>/dec64x_roundeven/g

```

```

1565 s\<roundevend64\>/dec64_roundeven/g
1566 s\<roundevenf128x\>/flt128x_roundeven/g
1567 s\<roundevenf128\>/flt128_roundeven/g
1568 s\<roundevenf16\>/flt16_roundeven/g
1569 s\<roundevenf32x\>/flt32x_roundeven/g
1570 s\<roundevenf32\>/flt32_roundeven/g
1571 s\<roundevenf64x\>/flt64x_roundeven/g
1572 s\<roundevenf64\>/flt64_roundeven/g
1573 s\<roundevenf\>/flt_roundeven/g
1574 s\<roundevenl\>/ldbl_roundeven/g
1575 # no change for roundeven
1576 s\<roundf128x\>/flt128x_round/g
1577 s\<roundf128\>/flt128_round/g
1578 s\<roundf16\>/flt16_round/g
1579 s\<roundf32x\>/flt32x_round/g
1580 s\<roundf32\>/flt32_round/g
1581 s\<roundf64x\>/flt64x_round/g
1582 s\<roundf64\>/flt64_round/g
1583 s\<rsqrtd128x\>/dec128x_rsqrtd/g
1584 s\<rsqrtd128\>/dec128_rsqrtd/g
1585 s\<rsqrtd32\>/dec32_rsqrtd/g
1586 s\<rsqrtd64x\>/dec64x_rsqrtd/g
1587 s\<rsqrtd64\>/dec64_rsqrtd/g
1588 s\<rsqrtdf128x\>/flt128x_rsqrtd/g
1589 s\<rsqrtdf128\>/flt128_rsqrtd/g
1590 s\<rsqrtdf16\>/flt16_rsqrtd/g
1591 s\<rsqrtdf32x\>/flt32x_rsqrtd/g
1592 s\<rsqrtdf32\>/flt32_rsqrtd/g
1593 s\<rsqrtdf64x\>/flt64x_rsqrtd/g
1594 s\<rsqrtdf64\>/flt64_rsqrtd/g
1595 s\<rsqrtdf\>/flt_rsqrtd/g
1596 s\<rsqrtdl\>/ldbl_rsqrtd/g
1597 # no change for rsqrtd
1598 s\<samequantumd128x\>/dec128x_samequantum/g
1599 s\<samequantumd128\>/dec128_samequantum/g
1600 s\<samequantumd32\>/dec32_samequantum/g
1601 s\<samequantumd64x\>/dec64x_samequantum/g
1602 s\<samequantumd64\>/dec64_samequantum/g
1603 s\<samequantumf128x\>/flt128x_samequantum/g
1604 s\<samequantumf128\>/flt128_samequantum/g
1605 s\<samequantumf16\>/flt16_samequantum/g
1606 s\<samequantumf32x\>/flt32x_samequantum/g
1607 s\<samequantumf32\>/flt32_samequantum/g
1608 s\<samequantumf64x\>/flt64x_samequantum/g
1609 s\<samequantumf64\>/flt64_samequantum/g
1610 # no change for samequantum
1611 s\<scalblnd128x\>/dec128x_scalbln/g
1612 s\<scalblnd128\>/dec128_scalbln/g
1613 s\<scalblnd32\>/dec32_scalbln/g
1614 s\<scalblnd64x\>/dec64x_scalbln/g
1615 s\<scalblnd64\>/dec64_scalbln/g
1616 s\<scalblnf128x\>/flt128x_scalbln/g
1617 s\<scalblnf128\>/flt128_scalbln/g
1618 s\<scalblnf16\>/flt16_scalbln/g
1619 s\<scalblnf32x\>/flt32x_scalbln/g
1620 s\<scalblnf32\>/flt32_scalbln/g
1621 s\<scalblnf64x\>/flt64x_scalbln/g
1622 s\<scalblnf64\>/flt64_scalbln/g
1623 s\<scalbnd128x\>/dec128x_scalbn/g
1624 s\<scalbnd128\>/dec128_scalbn/g
1625 s\<scalbnd32\>/dec32_scalbn/g
1626 s\<scalbnd64x\>/dec64x_scalbn/g
1627 s\<scalbnd64\>/dec64_scalbn/g

```

```
1628 s/\<scalbnf128x\>/flt128x_scalbn/g
1629 s/\<scalbnf128\>/flt128_scalbn/g
1630 s/\<scalbnf16\>/flt16_scalbn/g
1631 s/\<scalbnf32x\>/flt32x_scalbn/g
1632 s/\<scalbnf32\>/flt32_scalbn/g
1633 s/\<scalbnf64x\>/flt64x_scalbn/g
1634 s/\<scalbnf64\>/flt64_scalbn/g
1635 s/\<setpayloadd128x\>/dec128x_setpayload/g
1636 s/\<setpayloadd128\>/dec128_setpayload/g
1637 s/\<setpayloadd32\>/dec32_setpayload/g
1638 s/\<setpayloadd64x\>/dec64x_setpayload/g
1639 s/\<setpayloadd64\>/dec64_setpayload/g
1640 s/\<setpayloadf128x\>/flt128x_setpayload/g
1641 s/\<setpayloadf128\>/flt128_setpayload/g
1642 s/\<setpayloadf16\>/flt16_setpayload/g
1643 s/\<setpayloadf32x\>/flt32x_setpayload/g
1644 s/\<setpayloadf32\>/flt32_setpayload/g
1645 s/\<setpayloadf64x\>/flt64x_setpayload/g
1646 s/\<setpayloadf64\>/flt64_setpayload/g
1647 s/\<setpayloadf\>/flt_setpayload/g
1648 s/\<setpayloadl\>/ldbl_setpayload/g
1649 s/\<setpayloadsigd128x\>/dec128x_setpayloadsig/g
1650 s/\<setpayloadsigd128\>/dec128_setpayloadsig/g
1651 s/\<setpayloadsigd32\>/dec32_setpayloadsig/g
1652 s/\<setpayloadsigd64x\>/dec64x_setpayloadsig/g
1653 s/\<setpayloadsigd64\>/dec64_setpayloadsig/g
1654 s/\<setpayloadsigf128x\>/flt128x_setpayloadsig/g
1655 s/\<setpayloadsigf128\>/flt128_setpayloadsig/g
1656 s/\<setpayloadsigf16\>/flt16_setpayloadsig/g
1657 s/\<setpayloadsigf32x\>/flt32x_setpayloadsig/g
1658 s/\<setpayloadsigf32\>/flt32_setpayloadsig/g
1659 s/\<setpayloadsigf64x\>/flt64x_setpayloadsig/g
1660 s/\<setpayloadsigf64\>/flt64_setpayloadsig/g
1661 s/\<setpayloadsigf\>/flt_setpayloadsig/g
1662 s/\<setpayloadsigl\>/ldbl_setpayloadsig/g
1663 # no change for setpayloadsig
1664 # no change for setpayload
1665 s/\<sinhd128x\>/dec128x_sinh/g
1666 s/\<sinhd128\>/dec128_sinh/g
1667 s/\<sinhd32\>/dec32_sinh/g
1668 s/\<sinhd64x\>/dec64x_sinh/g
1669 s/\<sinhd64\>/dec64_sinh/g
1670 s/\<sinhf128x\>/flt128x_sinh/g
1671 s/\<sinhf128\>/flt128_sinh/g
1672 s/\<sinhf16\>/flt16_sinh/g
1673 s/\<sinhf32x\>/flt32x_sinh/g
1674 s/\<sinhf32\>/flt32_sinh/g
1675 s/\<sinhf64x\>/flt64x_sinh/g
1676 s/\<sinhf64\>/flt64_sinh/g
1677 s/\<sinhhd128x\>/dec128x_sinh/g
1678 s/\<sinhhd128\>/dec128_sinh/g
1679 s/\<sinhhd32\>/dec32_sinh/g
1680 s/\<sinhhd64x\>/dec64x_sinh/g
1681 s/\<sinhhd64\>/dec64_sinh/g
1682 s/\<sinhf128x\>/flt128x_sinh/g
1683 s/\<sinhf128\>/flt128_sinh/g
1684 s/\<sinhf16\>/flt16_sinh/g
1685 s/\<sinhf32x\>/flt32x_sinh/g
1686 s/\<sinhf32\>/flt32_sinh/g
1687 s/\<sinhf64x\>/flt64x_sinh/g
1688 s/\<sinhf64\>/flt64_sinh/g
1689 s/\<sinpid128x\>/dec128x_sinpi/g
1690 s/\<sinpid128\>/dec128_sinpi/g
```

```

1691 s\<sinpid32\>/dec32_sinpi/g
1692 s\<sinpid64x\>/dec64x_sinpi/g
1693 s\<sinpid64\>/dec64_sinpi/g
1694 s\<sinpif128x\>/flt128x_sinpi/g
1695 s\<sinpif128\>/flt128_sinpi/g
1696 s\<sinpif16\>/flt16_sinpi/g
1697 s\<sinpif32x\>/flt32x_sinpi/g
1698 s\<sinpif32\>/flt32_sinpi/g
1699 s\<sinpif64x\>/flt64x_sinpi/g
1700 s\<sinpif64\>/flt64_sinpi/g
1701 s\<sinpif\>/flt_sinpi/g
1702 s\<sinpil\>/ldbl_sinpi/g
1703 # no change for sinpi
1704 s\<sqrtd128x\>/dec128x_sqrt/g
1705 s\<sqrtd128\>/dec128_sqrt/g
1706 s\<sqrtd32\>/dec32_sqrt/g
1707 s\<sqrtd64x\>/dec64x_sqrt/g
1708 s\<sqrtd64\>/dec64_sqrt/g
1709 s\<sqrtf128x\>/flt128x_sqrt/g
1710 s\<sqrtf128\>/flt128_sqrt/g
1711 s\<sqrtf16\>/flt16_sqrt/g
1712 s\<sqrtf32x\>/flt32x_sqrt/g
1713 s\<sqrtf32\>/flt32_sqrt/g
1714 s\<sqrtf64x\>/flt64x_sqrt/g
1715 s\<sqrtf64\>/flt64_sqrt/g
1716 s\<tand128x\>/dec128x_tan/g
1717 s\<tand128\>/dec128_tan/g
1718 s\<tand32\>/dec32_tan/g
1719 s\<tand64x\>/dec64x_tan/g
1720 s\<tand64\>/dec64_tan/g
1721 s\<tanf128x\>/flt128x_tan/g
1722 s\<tanf128\>/flt128_tan/g
1723 s\<tanf16\>/flt16_tan/g
1724 s\<tanf32x\>/flt32x_tan/g
1725 s\<tanf32\>/flt32_tan/g
1726 s\<tanf64x\>/flt64x_tan/g
1727 s\<tanf64\>/flt64_tan/g
1728 s\<tanhd128x\>/dec128x_tanh/g
1729 s\<tanhd128\>/dec128_tanh/g
1730 s\<tanhd32\>/dec32_tanh/g
1731 s\<tanhd64x\>/dec64x_tanh/g
1732 s\<tanhd64\>/dec64_tanh/g
1733 s\<tanhf128x\>/flt128x_tanh/g
1734 s\<tanhf128\>/flt128_tanh/g
1735 s\<tanhf16\>/flt16_tanh/g
1736 s\<tanhf32x\>/flt32x_tanh/g
1737 s\<tanhf32\>/flt32_tanh/g
1738 s\<tanhf64x\>/flt64x_tanh/g
1739 s\<tanhf64\>/flt64_tanh/g
1740 s\<tanpid128x\>/dec128x_tanpi/g
1741 s\<tanpid128\>/dec128_tanpi/g
1742 s\<tanpid32\>/dec32_tanpi/g
1743 s\<tanpid64x\>/dec64x_tanpi/g
1744 s\<tanpid64\>/dec64_tanpi/g
1745 s\<tanpif128x\>/flt128x_tanpi/g
1746 s\<tanpif128\>/flt128_tanpi/g
1747 s\<tanpif16\>/flt16_tanpi/g
1748 s\<tanpif32x\>/flt32x_tanpi/g
1749 s\<tanpif32\>/flt32_tanpi/g
1750 s\<tanpif64x\>/flt64x_tanpi/g
1751 s\<tanpif64\>/flt64_tanpi/g
1752 s\<tanpif\>/flt_tanpi/g
1753 s\<tanpil\>/ldbl_tanpi/g

```

```

1754 # no change for tanpi
1755 s/^(<tgammad128x\>)/dec128x_tgamma/g
1756 s/^(<tgammad128\>)/dec128_tgamma/g
1757 s/^(<tgammad32\>)/dec32_tgamma/g
1758 s/^(<tgammad64x\>)/dec64x_tgamma/g
1759 s/^(<tgammad64\>)/dec64_tgamma/g
1760 s/^(<tgammaf128x\>)/flt128x_tgamma/g
1761 s/^(<tgammaf128\>)/flt128_tgamma/g
1762 s/^(<tgammaf16\>)/flt16_tgamma/g
1763 s/^(<tgammaf32x\>)/flt32x_tgamma/g
1764 s/^(<tgammaf32\>)/flt32_tgamma/g
1765 s/^(<tgammaf64x\>)/flt64x_tgamma/g
1766 s/^(<tgammaf64\>)/flt64_tgamma/g
1767 s/^(<truncd128x\>)/dec128x_trunc/g
1768 s/^(<truncd128\>)/dec128_trunc/g
1769 s/^(<truncd32\>)/dec32_trunc/g
1770 s/^(<truncd64x\>)/dec64x_trunc/g
1771 s/^(<truncd64\>)/dec64_trunc/g
1772 s/^(<truncf128x\>)/flt128x_trunc/g
1773 s/^(<truncf128\>)/flt128_trunc/g
1774 s/^(<truncf16\>)/flt16_trunc/g
1775 s/^(<truncf32x\>)/flt32x_trunc/g
1776 s/^(<truncf32\>)/flt32_trunc/g
1777 s/^(<truncf64x\>)/flt64x_trunc/g
1778 s/^(<truncf64\>)/flt64_trunc/g
1779 s/^(<ufromfpd128x\>)/dec128x_touint/g
1780 s/^(<ufromfpd128\>)/dec128_touint/g
1781 s/^(<ufromfpd32\>)/dec32_touint/g
1782 s/^(<ufromfpd64x\>)/dec64x_touint/g
1783 s/^(<ufromfpd64\>)/dec64_touint/g
1784 s/^(<ufromfpf128x\>)/flt128x_touint/g
1785 s/^(<ufromfpf128\>)/flt128_touint/g
1786 s/^(<ufromfpf16\>)/flt16_touint/g
1787 s/^(<ufromfpf32x\>)/flt32x_touint/g
1788 s/^(<ufromfpf32\>)/flt32_touint/g
1789 s/^(<ufromfpf64x\>)/flt64x_touint/g
1790 s/^(<ufromfpf64\>)/flt64_touint/g
1791 s/^(<ufromfpf\>)/flt_touint/g
1792 s/^(<ufromfpl\>)/ldbl_touint/g
1793 s/^(<ufromfpxd128x\>)/dec128x_touintx/g
1794 s/^(<ufromfpxd128\>)/dec128_touintx/g
1795 s/^(<ufromfpxd32\>)/dec32_touintx/g
1796 s/^(<ufromfpxd64x\>)/dec64x_touintx/g
1797 s/^(<ufromfpxd64\>)/dec64_touintx/g
1798 s/^(<ufromfpxf128x\>)/flt128x_touintx/g
1799 s/^(<ufromfpxf128\>)/flt128_touintx/g
1800 s/^(<ufromfpxf16\>)/flt16_touintx/g
1801 s/^(<ufromfpxf32x\>)/flt32x_touintx/g
1802 s/^(<ufromfpxf32\>)/flt32_touintx/g
1803 s/^(<ufromfpxf64x\>)/flt64x_touintx/g
1804 s/^(<ufromfpxf64\>)/flt64_touintx/g
1805 s/^(<ufromfpxf\>)/flt_touintx/g
1806 s/^(<ufromfpxl\>)/ldbl_touintx/g
1807 s/^(<ufromfpx\>)/touintx/g
1808 s/^(<ufromfp\>)/touint/g
1809 s/^(<_Decimal\([0-9]\{1,\}\)\_t\>)/dec\_1\_eval\_t/g
1810 s/^(<_Float\([0-9]\{1,\}\)\_t\>)/flt\_1\_eval\_t/g
1811 s/^(<_Decimal\([0-9]\{1,\}\[x]\{1,\}\)\>)/dec\_1\_t/g
1812 s/^(<_Float\([0-9]\{1,\}\[x]\{1,\}\)\>)/flt\_1\_t/g

```

## 6.2. An preprocessor header file for replacement

```

1 #define CR_DECIMAL_DIG FP_DECIMAL_DIG
2 #define FENV_DEC_ROUND FE_DEC_ROUND
3 #define FENV_ROUND FE_ROUND
4 #define HUGE_VAL_D128X D128X_HUGE_VAL
5 #define HUGE_VAL_D128 D128_HUGE_VAL
6 #define HUGE_VAL_D32 D32_HUGE_VAL
7 #define HUGE_VAL_D64X D64X_HUGE_VAL
8 #define HUGE_VAL_D64 D64_HUGE_VAL
9 #define HUGE_VAL_F128X F128X_HUGE_VAL
10 #define HUGE_VAL_F128 F128_HUGE_VAL
11 #define HUGE_VAL_F16 F16_HUGE_VAL
12 #define HUGE_VAL_F32X F32X_HUGE_VAL
13 #define HUGE_VAL_F32 F32_HUGE_VAL
14 #define HUGE_VAL_F64X F64X_HUGE_VAL
15 #define HUGE_VAL_F64 F64_HUGE_VAL
16 #define SNAND128X DEC128X_SNAN
17 #define SNAND128 DEC128_SNAN
18 #define SNAND32 DEC32_SNAN
19 #define SNAND64X DEC64X_SNAN
20 #define SNAND64 DEC64_SNAN
21 #define SNANF128X FLT128X_SNAN
22 #define SNANF128 FLT128_SNAN
23 #define SNANF16 FLT16_SNAN
24 #define SNANF32X FLT32X_SNAN
25 #define SNANF32 FLT32_SNAN
26 #define SNANF64X FLT64X_SNAN
27 #define SNANF64 FLT64_SNAN
28 #define SNANF FLT_SNAN
29 #define SNANL LDBL_SNAN
30 #define SNAN DBL_SNAN
31 #define acosd128x dec128x_acos
32 #define acosd128 dec128_acos
33 #define acosd32 dec32_acos
34 #define acosd64x dec64x_acos
35 #define acosd64 dec64_acos
36 #define acosf128x flt128x_acos
37 #define acosf128 flt128_acos
38 #define acosf16 flt16_acos
39 #define acosf32x flt32x_acos
40 #define acosf32 flt32_acos
41 #define acosf64x flt64x_acos
42 #define acosf64 flt64_acos
43 #define acoshd128x dec128x_acosh
44 #define acoshd128 dec128_acosh
45 #define acoshd32 dec32_acosh
46 #define acoshd64x dec64x_acosh
47 #define acoshd64 dec64_acosh
48 #define acoshf128x flt128x_acosh
49 #define acoshf128 flt128_acosh
50 #define acoshf16 flt16_acosh
51 #define acoshf32x flt32x_acosh
52 #define acoshf32 flt32_acosh
53 #define acoshf64x flt64x_acosh
54 #define acoshf64 flt64_acosh
55 #define acospid128x dec128x_acospi
56 #define acospid128 dec128_acospi
57 #define acospid32 dec32_acospi
58 #define acospid64x dec64x_acospi
59 #define acospid64 dec64_acospi
60 #define acospi128x flt128x_acospi
61 #define acospi128 flt128_acospi

```

```
62 #define acospi16 flt16_acospi
63 #define acospi32x flt32x_acospi
64 #define acospi32 flt32_acospi
65 #define acospi64x flt64x_acospi
66 #define acospi64 flt64_acospi
67 #define acospi flt_acospi
68 #define acospi1 ldbl_acospi
69 #define asind128x dec128x_asin
70 #define asind128 dec128_asin
71 #define asind32 dec32_asin
72 #define asind64x dec64x_asin
73 #define asind64 dec64_asin
74 #define asinf128x flt128x_asin
75 #define asinf128 flt128_asin
76 #define asinf16 flt16_asin
77 #define asinf32x flt32x_asin
78 #define asinf32 flt32_asin
79 #define asinf64x flt64x_asin
80 #define asinf64 flt64_asin
81 #define asinhd128x dec128x_asinh
82 #define asinhd128 dec128_asinh
83 #define asinhd32 dec32_asinh
84 #define asinhd64x dec64x_asinh
85 #define asinhd64 dec64_asinh
86 #define asinhf128x flt128x_asinh
87 #define asinhf128 flt128_asinh
88 #define asinhf16 flt16_asinh
89 #define asinhf32x flt32x_asinh
90 #define asinhf32 flt32_asinh
91 #define asinhf64x flt64x_asinh
92 #define asinhf64 flt64_asinh
93 #define asinpid128x dec128x_asinpi
94 #define asinpid128 dec128_asinpi
95 #define asinpid32 dec32_asinpi
96 #define asinpid64x dec64x_asinpi
97 #define asinpid64 dec64_asinpi
98 #define asinpif128x flt128x_asinpi
99 #define asinpif128 flt128_asinpi
100 #define asinpif16 flt16_asinpi
101 #define asinpif32x flt32x_asinpi
102 #define asinpif32 flt32_asinpi
103 #define asinpif64x flt64x_asinpi
104 #define asinpif64 flt64_asinpi
105 #define asinpif flt_asinpi
106 #define asinpil ldbl_asinpi
107 #define atan2d128x dec128x_atan2
108 #define atan2d128 dec128_atan2
109 #define atan2d32 dec32_atan2
110 #define atan2d64x dec64x_atan2
111 #define atan2d64 dec64_atan2
112 #define atan2f128x flt128x_atan2
113 #define atan2f128 flt128_atan2
114 #define atan2f32x flt32x_atan2
115 #define atan2f32 flt32_atan2
116 #define atan2f64x flt64x_atan2
117 #define atan2f64 flt64_atan2
118 #define atan2pid128x dec128x_atan2pi
119 #define atan2pid128 dec128_atan2pi
120 #define atan2pid32 dec32_atan2pi
121 #define atan2pid64x dec64x_atan2pi
122 #define atan2pid64 dec64_atan2pi
123 #define atan2pif128x flt128x_atan2pi
124 #define atan2pif128 flt128_atan2pi
```

```
125 #define atan2pif32x flt32x_atan2pi
126 #define atan2pif32 flt32_atan2pi
127 #define atan2pif64x flt64x_atan2pi
128 #define atan2pif64 flt64_atan2pi
129 #define atan2pif flt_atan2pi
130 #define atan2pil ldbl_atan2pi
131 #define atand128x dec128x_atan
132 #define atand128 dec128_atan
133 #define atand32 dec32_atan
134 #define atand64x dec64x_atan
135 #define atand64 dec64_atan
136 #define atanf128x flt128x_atan
137 #define atanf128 flt128_atan
138 #define atanf16 flt16_atan
139 #define atanf32x flt32x_atan
140 #define atanf32 flt32_atan
141 #define atanf64x flt64x_atan
142 #define atanf64 flt64_atan
143 #define atanhd128x dec128x_atanh
144 #define atanhd128 dec128_atanh
145 #define atanhd32 dec32_atanh
146 #define atanhd64x dec64x_atanh
147 #define atanhd64 dec64_atanh
148 #define atanhf128x flt128x_atanh
149 #define atanhf128 flt128_atanh
150 #define atanhf16 flt16_atanh
151 #define atanhf32x flt32x_atanh
152 #define atanhf32 flt32_atanh
153 #define atanhf64x flt64x_atanh
154 #define atanhf64 flt64_atanh
155 #define atanpid128x dec128x_atanpi
156 #define atanpid128 dec128_atanpi
157 #define atanpid32 dec32_atanpi
158 #define atanpid64x dec64x_atanpi
159 #define atanpid64 dec64_atanpi
160 #define atanpif128x flt128x_atanpi
161 #define atanpif128 flt128_atanpi
162 #define atanpif16 flt16_atanpi
163 #define atanpif32x flt32x_atanpi
164 #define atanpif32 flt32_atanpi
165 #define atanpif64x flt64x_atanpi
166 #define atanpif64 flt64_atanpi
167 #define atanpif flt_atanpi
168 #define atanpil ldbl_atanpi
169 #define cabsf128x flt128x_cabs
170 #define cabsf128 flt128_cabs
171 #define cabsf32x flt32x_cabs
172 #define cabsf32 flt32_cabs
173 #define cabsf64x flt64x_cabs
174 #define cabsf64 flt64_cabs
175 #define cacoshf128x flt128x_cacosh
176 #define cacoshf128 flt128_cacosh
177 #define cacoshf32x flt32x_cacosh
178 #define cacoshf32 flt32_cacosh
179 #define cacoshf64x flt64x_cacosh
180 #define cacoshf64 flt64_cacosh
181 #define cacoshf128x flt128x_cacosh
182 #define cacoshf128 flt128_cacosh
183 #define cacoshf32x flt32x_cacosh
184 #define cacoshf32 flt32_cacosh
185 #define cacoshf64x flt64x_cacosh
186 #define cacoshf64 flt64_cacosh
187 #define canonicalized128x dec128x_canonicalize
```



```
188 #define canonicalized128 dec128_canonicalize
189 #define canonicalized32 dec32_canonicalize
190 #define canonicalized64x dec64x_canonicalize
191 #define canonicalized64 dec64_canonicalize
192 #define canonicalizef128x flt128x_canonicalize
193 #define canonicalizef128 flt128_canonicalize
194 #define canonicalizef16 flt16_canonicalize
195 #define canonicalizef32x flt32x_canonicalize
196 #define canonicalizef32 flt32_canonicalize
197 #define canonicalizef64x flt64x_canonicalize
198 #define canonicalizef64 flt64_canonicalize
199 #define canonicalizef flt_canonicalize
200 #define canonicalize1 ldbl_canonicalize
201 #define cargf128x flt128x_carg
202 #define cargf128 flt128_carg
203 #define cargf32x flt32x_carg
204 #define cargf32 flt32_carg
205 #define cargf64x flt64x_carg
206 #define cargf64 flt64_carg
207 #define casin128x flt128x_casin
208 #define casin128 flt128_casin
209 #define casin32x flt32x_casin
210 #define casin32 flt32_casin
211 #define casin64x flt64x_casin
212 #define casin64 flt64_casin
213 #define casinhf128x flt128x_casinh
214 #define casinhf128 flt128_casinh
215 #define casinhf32x flt32x_casinh
216 #define casinhf32 flt32_casinh
217 #define casinhf64x flt64x_casinh
218 #define casinhf64 flt64_casinh
219 #define catanf128x flt128x_catan
220 #define catanf128 flt128_catan
221 #define catanf32x flt32x_catan
222 #define catanf32 flt32_catan
223 #define catanf64x flt64x_catan
224 #define catanf64 flt64_catan
225 #define catanhf128x flt128x_catanh
226 #define catanhf128 flt128_catanh
227 #define catanhf32x flt32x_catanh
228 #define catanhf32 flt32_catanh
229 #define catanhf64x flt64x_catanh
230 #define catanhf64 flt64_catanh
231 #define cbrtd128x dec128x_cbirt
232 #define cbrtd128 dec128_cbirt
233 #define cbrtd32 dec32_cbirt
234 #define cbrtd64x dec64x_cbirt
235 #define cbrtd64 dec64_cbirt
236 #define cbrtf128x flt128x_cbirt
237 #define cbrtf128 flt128_cbirt
238 #define cbrtf16 flt16_cbirt
239 #define cbrtf32x flt32x_cbirt
240 #define cbrtf32 flt32_cbirt
241 #define cbrtf64x flt64x_cbirt
242 #define cbrtf64 flt64_cbirt
243 #define ccosf128x flt128x_ccos
244 #define ccosf128 flt128_ccos
245 #define ccosf32x flt32x_ccos
246 #define ccosf32 flt32_ccos
247 #define ccosf64x flt64x_ccos
248 #define ccosf64 flt64_ccos
249 #define ccoshf128x flt128x_ccosh
250 #define ccoshf128 flt128_ccosh
```

```
251 #define ccoshf32x flt32x_ccosh
252 #define ccoshf32 flt32_ccosh
253 #define ccoshf64x flt64x_ccosh
254 #define ccoshf64 flt64_ccosh
255 #define ceild128x dec128x_ceil
256 #define ceild128 dec128_ceil
257 #define ceild32 dec32_ceil
258 #define ceild64x dec64x_ceil
259 #define ceild64 dec64_ceil
260 #define ceilf128x flt128x_ceil
261 #define ceilf128 flt128_ceil
262 #define ceilf16 flt16_ceil
263 #define ceilf32x flt32x_ceil
264 #define ceilf32 flt32_ceil
265 #define ceilf64x flt64x_ceil
266 #define ceilf64 flt64_ceil
267 #define cexpf128x flt128x_cexp
268 #define cexpf128 flt128_cexp
269 #define cexpf32x flt32x_cexp
270 #define cexpf32 flt32_cexp
271 #define cexpf64x flt64x_cexp
272 #define cexpf64 flt64_cexp
273 #define cimagf128x flt128x_cimag
274 #define cimagf128 flt128_cimag
275 #define cimagf32x flt32x_cimag
276 #define cimagf32 flt32_cimag
277 #define cimagf64x flt64x_cimag
278 #define cimagf64 flt64_cimag
279 #define clogf128x flt128x_clog
280 #define clogf128 flt128_clog
281 #define clogf32x flt32x_clog
282 #define clogf32 flt32_clog
283 #define clogf64x flt64x_clog
284 #define clogf64 flt64_clog
285 #define compoundnd128x dec128x_compoundn
286 #define compoundnd128 dec128_compoundn
287 #define compoundnd32 dec32_compoundn
288 #define compoundnd64x dec64x_compoundn
289 #define compoundnd64 dec64_compoundn
290 #define compoundnf128x flt128x_compoundn
291 #define compoundnf128 flt128_compoundn
292 #define compoundnf16 flt16_compoundn
293 #define compoundnf32x flt32x_compoundn
294 #define compoundnf32 flt32_compoundn
295 #define compoundnf64x flt64x_compoundn
296 #define compoundnf64 flt64_compoundn
297 #define compoundnf flt_compoundn
298 #define compoundn1 ldbl_compoundn
299 #define conjf128x flt128x_conj
300 #define conjf128 flt128_conj
301 #define conjf32x flt32x_conj
302 #define conjf32 flt32_conj
303 #define conjf64x flt64x_conj
304 #define conjf64 flt64_conj
305 #define copysignd128x dec128x_copysign
306 #define copysignd128 dec128_copysign
307 #define copysignd32 dec32_copysign
308 #define copysignd64x dec64x_copysign
309 #define copysignd64 dec64_copysign
310 #define copysignf128x flt128x_copysign
311 #define copysignf128 flt128_copysign
312 #define copysignf16 flt16_copysign
313 #define copysignf32x flt32x_copysign
```

```
314 #define copysignf32 flt32_copysign
315 #define copysignf64x flt64x_copysign
316 #define copysignf64 flt64_copysign
317 #define cosd128x dec128x_cos
318 #define cosd128 dec128_cos
319 #define cosd32 dec32_cos
320 #define cosd64x dec64x_cos
321 #define cosd64 dec64_cos
322 #define cosf128x flt128x_cos
323 #define cosf128 flt128_cos
324 #define cosf16 flt16_cos
325 #define cosf32x flt32x_cos
326 #define cosf32 flt32_cos
327 #define cosf64x flt64x_cos
328 #define cosf64 flt64_cos
329 #define coshd128x dec128x_cosh
330 #define coshd128 dec128_cosh
331 #define coshd32 dec32_cosh
332 #define coshd64x dec64x_cosh
333 #define coshd64 dec64_cosh
334 #define coshf128x flt128x_cosh
335 #define coshf128 flt128_cosh
336 #define coshf16 flt16_cosh
337 #define coshf32x flt32x_cosh
338 #define coshf32 flt32_cosh
339 #define coshf64x flt64x_cosh
340 #define coshf64 flt64_cosh
341 #define cospid128x dec128x_cospi
342 #define cospid128 dec128_cospi
343 #define cospid32 dec32_cospi
344 #define cospid64x dec64x_cospi
345 #define cospid64 dec64_cospi
346 #define cospif128x flt128x_cospi
347 #define cospif128 flt128_cospi
348 #define cospif16 flt16_cospi
349 #define cospif32x flt32x_cospi
350 #define cospif32 flt32_cospi
351 #define cospif64x flt64x_cospi
352 #define cospif64 flt64_cospi
353 #define cospif flt_cospi
354 #define cospil ldbl_cospi
355 #define cpowf128x flt128x_cpow
356 #define cpowf128 flt128_cpow
357 #define cpowf32x flt32x_cpow
358 #define cpowf32 flt32_cpow
359 #define cpowf64x flt64x_cpow
360 #define cpowf64 flt64_cpow
361 #define cprojf128x flt128x_cproj
362 #define cprojf128 flt128_cproj
363 #define cprojf32x flt32x_cproj
364 #define cprojf32 flt32_cproj
365 #define cprojf64x flt64x_cproj
366 #define cprojf64 flt64_cproj
367 #define crealf128x flt128x_creal
368 #define crealf128 flt128_creal
369 #define crealf32x flt32x_creal
370 #define crealf32 flt32_creal
371 #define crealf64x flt64x_creal
372 #define crealf64 flt64_creal
373 #define csinf128x flt128x_csin
374 #define csinf128 flt128_csin
375 #define csinf32x flt32x_csin
376 #define csinf32 flt32_csin
```

```
377 #define csinf64x flt64x_csin
378 #define csinf64 flt64_csin
379 #define csinhf128x flt128x_csinh
380 #define csinhf128 flt128_csinh
381 #define csinhf32x flt32x_csinh
382 #define csinhf32 flt32_csinh
383 #define csinhf64x flt64x_csinh
384 #define csinhf64 flt64_csinh
385 #define csqrtf128x flt128x_csqrt
386 #define csqrtf128 flt128_csqrt
387 #define csqrtf32x flt32x_csqrt
388 #define csqrtf32 flt32_csqrt
389 #define csqrtf64x flt64x_csqrt
390 #define csqrtf64 flt64_csqrt
391 #define ctanf128x flt128x_ctan
392 #define ctanf128 flt128_ctan
393 #define ctanf32x flt32x_ctan
394 #define ctanf32 flt32_ctan
395 #define ctanf64x flt64x_ctan
396 #define ctanf64 flt64_ctan
397 #define ctanhf128x flt128x_ctanh
398 #define ctanhf128 flt128_ctanh
399 #define ctanhf32x flt32x_ctanh
400 #define ctanhf32 flt32_ctanh
401 #define ctanhf64x flt64x_ctanh
402 #define ctanhf64 flt64_ctanh
403 #define d128add128x dec128x_todec128_add
404 #define d128add todec128_add
405 #define d128div128x dec128x_todec128_div
406 #define d128div todec128_div
407 #define d128fmad128x dec128x_todec128_fma
408 #define d128fma todec128_fma
409 #define d128muld128x dec128x_todec128_mul
410 #define d128mul todec128_mul
411 #define d128sqrtd128x dec128x_todec128_sqrt
412 #define d128sqrt todec128_sqrt
413 #define d128subd128x dec128x_todec128_sub
414 #define d128sub todec128_sub
415 #define d32add128x dec128x_todec32_add
416 #define d32add128 dec128_todec32_add
417 #define d32add64x dec64x_todec32_add
418 #define d32add64 dec64_todec32_add
419 #define d32add todec32_add
420 #define d32divd128x dec128x_todec32_div
421 #define d32divd128 dec128_todec32_div
422 #define d32divd64x dec64x_todec32_div
423 #define d32divd64 dec64_todec32_div
424 #define d32div todec32_div
425 #define d32fmad128x dec128x_todec32_fma
426 #define d32fmad128 dec128_todec32_fma
427 #define d32fmad64x dec64x_todec32_fma
428 #define d32fmad64 dec64_todec32_fma
429 #define d32fma todec32_fma
430 #define d32muld128x dec128x_todec32_mul
431 #define d32muld128 dec128_todec32_mul
432 #define d32muld64x dec64x_todec32_mul
433 #define d32muld64 dec64_todec32_mul
434 #define d32mul todec32_mul
435 #define d32sqrtd128x dec128x_todec32_sqrt
436 #define d32sqrtd128 dec128_todec32_sqrt
437 #define d32sqrtd64x dec64x_todec32_sqrt
438 #define d32sqrtd64 dec64_todec32_sqrt
439 #define d32sqrt todec32_sqrt
```

```
440 #define d32subd128x dec128x_todec32_sub
441 #define d32subd128 dec128_todec32_sub
442 #define d32subd64x dec64x_todec32_sub
443 #define d32subd64 dec64_todec32_sub
444 #define d32sub todec32_sub
445 #define d64addd128x dec128x_todec64_add
446 #define d64addd128 dec128_todec64_add
447 #define d64addd64x dec64x_todec64_add
448 #define d64add todec64_add
449 #define d64divd128x dec128x_todec64_div
450 #define d64divd128 dec128_todec64_div
451 #define d64divd64x dec64x_todec64_div
452 #define d64div todec64_div
453 #define d64fmad128x dec128x_todec64_fma
454 #define d64fmad128 dec128_todec64_fma
455 #define d64fmad64x dec64x_todec64_fma
456 #define d64fma todec64_fma
457 #define d64muld128x dec128x_todec64_mul
458 #define d64muld128 dec128_todec64_mul
459 #define d64muld64x dec64x_todec64_mul
460 #define d64mul todec64_mul
461 #define d64sqrtd128x dec128x_todec64_sqrt
462 #define d64sqrtd128 dec128_todec64_sqrt
463 #define d64sqrtd64x dec64x_todec64_sqrt
464 #define d64sqrt todec64_sqrt
465 #define d64subd128x dec128x_todec64_sub
466 #define d64subd128 dec128_todec64_sub
467 #define d64subd64x dec64x_todec64_sub
468 #define d64sub todec64_sub
469 #define d64xaddd128x dec128x_todec64x_add
470 #define d64xaddd128 dec128_todec64x_add
471 #define d64xadd todec64x_add
472 #define d64xdivd128x dec128x_todec64x_div
473 #define d64xdivd128 dec128_todec64x_div
474 #define d64xdiv todec64x_div
475 #define d64xfmad128x dec128x_todec64x_fma
476 #define d64xfmad128 dec128_todec64x_fma
477 #define d64xfma todec64x_fma
478 #define d64xmuld128x dec128x_todec64x_mul
479 #define d64xmuld128 dec128_todec64x_mul
480 #define d64xmul todec64x_mul
481 #define d64xsqrtd128x dec128x_todec64x_sqrt
482 #define d64xsqrtd128 dec128_todec64x_sqrt
483 #define d64xsqrt todec64x_sqrt
484 #define d64xsubd128x dec128x_todec64x_sub
485 #define d64xsubd128 dec128_todec64x_sub
486 #define d64xsub todec64x_sub
487 #define daddl ldbl_todbl_add
488 #define dadd todbl_add
489 #define ddivl ldbl_todbl_div
490 #define ddiv todbl_div
491 #define decodebind128x dec128x_decodebin
492 #define decodebind128 dec128_decodebin
493 #define decodebind32 dec32_decodebin
494 #define decodebind64x dec64x_decodebin
495 #define decodebind64 dec64_decodebin
496 #define decodedecd128x dec128x_decodedec
497 #define decodedecd128 dec128_decodedec
498 #define decodedecd32 dec32_decodedec
499 #define decodedecd64x dec64x_decodedec
500 #define decodedecd64 dec64_decodedec
501 #define dfmal ldbl_todbl_fma
502 #define dfma todbl_fma
```

```
503 #define dmull ldbl_todbl_mul
504 #define dmul todbl_mul
505 #define dsqrtl ldbl_todbl_sqrt
506 #define dsqrt todbl_sqrt
507 #define dsubl ldbl_todbl_sub
508 #define dsub todbl_sub
509 #define encodebind128x dec128x_encodebin
510 #define encodebind128 dec128_encodebin
511 #define encodebind32 dec32_encodebin
512 #define encodebind64x dec64x_encodebin
513 #define encodebind64 dec64_encodebin
514 #define encodedecd128x dec128x_encodedec
515 #define encodedecd128 dec128_encodedec
516 #define encodedecd32 dec32_encodedec
517 #define encodedecd64x dec64x_encodedec
518 #define encodedecd64 dec64_encodedec
519 #define erfcd128x dec128x_erfc
520 #define erfcd128 dec128_erfc
521 #define erfcd32 dec32_erfc
522 #define erfcd64x dec64x_erfc
523 #define erfcd64 dec64_erfc
524 #define erfcf128x flt128x_erfc
525 #define erfcf128 flt128_erfc
526 #define erfcf16 flt16_erfc
527 #define erfcf32x flt32x_erfc
528 #define erfcf32 flt32_erfc
529 #define erfcf64x flt64x_erfc
530 #define erfcf64 flt64_erfc
531 #define erfd128x dec128x_erf
532 #define erfd128 dec128_erf
533 #define erfd32 dec32_erf
534 #define erfd64x dec64x_erf
535 #define erfd64 dec64_erf
536 #define erff128x flt128x_erf
537 #define erff128 flt128_erf
538 #define erff16 flt16_erf
539 #define erff32x flt32x_erf
540 #define erff32 flt32_erf
541 #define erff64x flt64x_erf
542 #define erff64 flt64_erf
543 #define exp10d128x dec128x_exp10
544 #define exp10d128 dec128_exp10
545 #define exp10d32 dec32_exp10
546 #define exp10d64x dec64x_exp10
547 #define exp10d64 dec64_exp10
548 #define exp10f128x flt128x_exp10
549 #define exp10f128 flt128_exp10
550 #define exp10f32x flt32x_exp10
551 #define exp10f32 flt32_exp10
552 #define exp10f64x flt64x_exp10
553 #define exp10f64 flt64_exp10
554 #define exp10f flt_exp10
555 #define exp10l ldbl_exp10
556 #define exp10m1d128x dec128x_exp10m1
557 #define exp10m1d128 dec128_exp10m1
558 #define exp10m1d32 dec32_exp10m1
559 #define exp10m1d64x dec64x_exp10m1
560 #define exp10m1d64 dec64_exp10m1
561 #define exp10m1f128x flt128x_exp10m1
562 #define exp10m1f128 flt128_exp10m1
563 #define exp10m1f32x flt32x_exp10m1
564 #define exp10m1f32 flt32_exp10m1
565 #define exp10m1f64x flt64x_exp10m1
```

```
566 #define exp10m1f64 flt64_exp10m1
567 #define exp10m1f flt_exp10m1
568 #define exp10m1l ldbl_exp10m1
569 #define exp2d128x dec128x_exp2
570 #define exp2d128 dec128_exp2
571 #define exp2d32 dec32_exp2
572 #define exp2d64x dec64x_exp2
573 #define exp2d64 dec64_exp2
574 #define exp2f128x flt128x_exp2
575 #define exp2f128 flt128_exp2
576 #define exp2f32x flt32x_exp2
577 #define exp2f32 flt32_exp2
578 #define exp2f64x flt64x_exp2
579 #define exp2f64 flt64_exp2
580 #define exp2m1d128x dec128x_exp2m1
581 #define exp2m1d128 dec128_exp2m1
582 #define exp2m1d32 dec32_exp2m1
583 #define exp2m1d64x dec64x_exp2m1
584 #define exp2m1d64 dec64_exp2m1
585 #define exp2m1f128x flt128x_exp2m1
586 #define exp2m1f128 flt128_exp2m1
587 #define exp2m1f32x flt32x_exp2m1
588 #define exp2m1f32 flt32_exp2m1
589 #define exp2m1f64x flt64x_exp2m1
590 #define exp2m1f64 flt64_exp2m1
591 #define exp2m1f flt_exp2m1
592 #define exp2m1l ldbl_exp2m1
593 #define expd128x dec128x_exp
594 #define expd128 dec128_exp
595 #define expd32 dec32_exp
596 #define expd64x dec64x_exp
597 #define expd64 dec64_exp
598 #define expf128x flt128x_exp
599 #define expf128 flt128_exp
600 #define expf16 flt16_exp
601 #define expf32x flt32x_exp
602 #define expf32 flt32_exp
603 #define expf64x flt64x_exp
604 #define expf64 flt64_exp
605 #define expm1d128x dec128x_expm1
606 #define expm1d128 dec128_expm1
607 #define expm1d32 dec32_expm1
608 #define expm1d64x dec64x_expm1
609 #define expm1d64 dec64_expm1
610 #define expm1f128x flt128x_expm1
611 #define expm1f128 flt128_expm1
612 #define expm1f32x flt32x_expm1
613 #define expm1f32 flt32_expm1
614 #define expm1f64x flt64x_expm1
615 #define expm1f64 flt64_expm1
616 #define f128addf128x flt128x_toflt128_add
617 #define f128add toflt128_add
618 #define f128divf128x flt128x_toflt128_div
619 #define f128div toflt128_div
620 #define f128faf128x flt128x_toflt128_fma
621 #define f128fma toflt128_fma
622 #define f128mulf128x flt128x_toflt128_mul
623 #define f128mul toflt128_mul
624 #define f128sqrtf128x flt128x_toflt128_sqrt
625 #define f128sqrt toflt128_sqrt
626 #define f128subf128x flt128x_toflt128_sub
627 #define f128sub toflt128_sub
628 #define f16addf128x flt128x_toflt16_add
```

```
629 #define f16addf128 flt128_toflt16_add
630 #define f16addf32x flt32x_toflt16_add
631 #define f16addf32 flt32_toflt16_add
632 #define f16addf64x flt64x_toflt16_add
633 #define f16addf64 flt64_toflt16_add
634 #define f16add toflt16_add
635 #define f16divf128x flt128x_toflt16_div
636 #define f16divf128 flt128_toflt16_div
637 #define f16divf32x flt32x_toflt16_div
638 #define f16divf32 flt32_toflt16_div
639 #define f16divf64x flt64x_toflt16_div
640 #define f16divf64 flt64_toflt16_div
641 #define f16div toflt16_div
642 #define f16maf128x flt128x_toflt16_fma
643 #define f16maf128 flt128_toflt16_fma
644 #define f16maf32x flt32x_toflt16_fma
645 #define f16maf32 flt32_toflt16_fma
646 #define f16maf64x flt64x_toflt16_fma
647 #define f16maf64 flt64_toflt16_fma
648 #define f16fma toflt16_fma
649 #define f16mulf128x flt128x_toflt16_mul
650 #define f16mulf128 flt128_toflt16_mul
651 #define f16mulf32x flt32x_toflt16_mul
652 #define f16mulf32 flt32_toflt16_mul
653 #define f16mulf64x flt64x_toflt16_mul
654 #define f16mulf64 flt64_toflt16_mul
655 #define f16mul toflt16_mul
656 #define f16sqrtf128x flt128x_toflt16_sqrt
657 #define f16sqrtf128 flt128_toflt16_sqrt
658 #define f16sqrtf32x flt32x_toflt16_sqrt
659 #define f16sqrtf32 flt32_toflt16_sqrt
660 #define f16sqrtf64x flt64x_toflt16_sqrt
661 #define f16sqrtf64 flt64_toflt16_sqrt
662 #define f16sqrt toflt16_sqrt
663 #define f16subf128x flt128x_toflt16_sub
664 #define f16subf128 flt128_toflt16_sub
665 #define f16subf32x flt32x_toflt16_sub
666 #define f16subf32 flt32_toflt16_sub
667 #define f16subf64x flt64x_toflt16_sub
668 #define f16subf64 flt64_toflt16_sub
669 #define f16sub toflt16_sub
670 #define f32addf128x flt128x_toflt32_add
671 #define f32addf128 flt128_toflt32_add
672 #define f32addf32x flt32x_toflt32_add
673 #define f32addf64x flt64x_toflt32_add
674 #define f32addf64 flt64_toflt32_add
675 #define f32add toflt32_add
676 #define f32divf128x flt128x_toflt32_div
677 #define f32divf128 flt128_toflt32_div
678 #define f32divf32x flt32x_toflt32_div
679 #define f32divf64x flt64x_toflt32_div
680 #define f32divf64 flt64_toflt32_div
681 #define f32div toflt32_div
682 #define f32maf128x flt128x_toflt32_fma
683 #define f32maf128 flt128_toflt32_fma
684 #define f32maf32x flt32x_toflt32_fma
685 #define f32maf64x flt64x_toflt32_fma
686 #define f32maf64 flt64_toflt32_fma
687 #define f32fma toflt32_fma
688 #define f32mulf128x flt128x_toflt32_mul
689 #define f32mulf128 flt128_toflt32_mul
690 #define f32mulf32x flt32x_toflt32_mul
691 #define f32mulf64x flt64x_toflt32_mul
```



```
692 #define f32mulf64 flt64_toflt32_mul
693 #define f32mul toflt32_mul
694 #define f32sqrtf128x flt128x_toflt32_sqrt
695 #define f32sqrtf128 flt128_toflt32_sqrt
696 #define f32sqrtf32x flt32x_toflt32_sqrt
697 #define f32sqrtf64x flt64x_toflt32_sqrt
698 #define f32sqrtf64 flt64_toflt32_sqrt
699 #define f32sqrt toflt32_sqrt
700 #define f32subf128x flt128x_toflt32_sub
701 #define f32subf128 flt128_toflt32_sub
702 #define f32subf32x flt32x_toflt32_sub
703 #define f32subf64x flt64x_toflt32_sub
704 #define f32subf64 flt64_toflt32_sub
705 #define f32sub toflt32_sub
706 #define f32xaddf128x flt128x_toflt32x_add
707 #define f32xaddf128 flt128_toflt32x_add
708 #define f32xaddf64x flt64x_toflt32x_add
709 #define f32xaddf64 flt64_toflt32x_add
710 #define f32xadd toflt32x_add
711 #define f32xdivf128x flt128x_toflt32x_div
712 #define f32xdivf128 flt128_toflt32x_div
713 #define f32xdivf64x flt64x_toflt32x_div
714 #define f32xdivf64 flt64_toflt32x_div
715 #define f32xdiv toflt32x_div
716 #define f32xfmaf128x flt128x_toflt32x_fma
717 #define f32xfmaf128 flt128_toflt32x_fma
718 #define f32xfmaf64x flt64x_toflt32x_fma
719 #define f32xfmaf64 flt64_toflt32x_fma
720 #define f32xfma toflt32x_fma
721 #define f32xmulf128x flt128x_toflt32x_mul
722 #define f32xmulf128 flt128_toflt32x_mul
723 #define f32xmulf64x flt64x_toflt32x_mul
724 #define f32xmulf64 flt64_toflt32x_mul
725 #define f32xmul toflt32x_mul
726 #define f32xsqrtf128x flt128x_toflt32x_sqrt
727 #define f32xsqrtf128 flt128_toflt32x_sqrt
728 #define f32xsqrtf64x flt64x_toflt32x_sqrt
729 #define f32xsqrtf64 flt64_toflt32x_sqrt
730 #define f32xsqrt toflt32x_sqrt
731 #define f32xsubf128x flt128x_toflt32x_sub
732 #define f32xsubf128 flt128_toflt32x_sub
733 #define f32xsubf64x flt64x_toflt32x_sub
734 #define f32xsubf64 flt64_toflt32x_sub
735 #define f32xsub toflt32x_sub
736 #define f64addf128x flt128x_toflt64_add
737 #define f64addf64x flt64x_toflt64_add
738 #define f64add toflt64_add
739 #define f64divf128x flt128x_toflt64_div
740 #define f64divf64x flt64x_toflt64_div
741 #define f64div toflt64_div
742 #define f64fmaf128x flt128x_toflt64_fma
743 #define f64fmaf64x flt64x_toflt64_fma
744 #define f64fma toflt64_fma
745 #define f64mulf128x flt128x_toflt64_mul
746 #define f64mulf64x flt64x_toflt64_mul
747 #define f64mul toflt64_mul
748 #define f64sqrtf128x flt128x_toflt64_sqrt
749 #define f64sqrtf64x flt64x_toflt64_sqrt
750 #define f64sqrt toflt64_sqrt
751 #define f64subf128x flt128x_toflt64_sub
752 #define f64subf64x flt64x_toflt64_sub
753 #define f64sub toflt64_sub
754 #define f64xaddf128x flt128x_toflt64x_add
```

```
755 #define f64xadd toflt64x_add
756 #define f64xdivf128x flt128x_toflt64x_div
757 #define f64xdiv toflt64x_div
758 #define f64xfmaf128x flt128x_toflt64x_fma
759 #define f64xfma toflt64x_fma
760 #define f64xmulf128x flt128x_toflt64x_mul
761 #define f64xmul toflt64x_mul
762 #define f64xsqrtf128x flt128x_toflt64x_sqrt
763 #define f64xsqrt toflt64x_sqrt
764 #define f64xsubf128x flt128x_toflt64x_sub
765 #define f64xsub toflt64x_sub
766 #define fabsd128x dec128x_fabs
767 #define fabsd128 dec128_fabs
768 #define fabsd32 dec32_fabs
769 #define fabsd64x dec64x_fabs
770 #define fabsd64 dec64_fabs
771 #define fabsf128x flt128x_fabs
772 #define fabsf128 flt128_fabs
773 #define fabsf16 flt16_fabs
774 #define fabsf32x flt32x_fabs
775 #define fabsf32 flt32_fabs
776 #define fabsf64x flt64x_fabs
777 #define fabsf64 flt64_fabs
778 #define faddl ldbl_toflt_add
779 #define fadd toflt_add
780 #define fdimd128x dec128x_fdim
781 #define fdimd128 dec128_fdim
782 #define fdimd32 dec32_fdim
783 #define fdimd64x dec64x_fdim
784 #define fdimd64 dec64_fdim
785 #define fdimf128x flt128x_fdim
786 #define fdimf128 flt128_fdim
787 #define fdimf16 flt16_fdim
788 #define fdimf32x flt32x_fdim
789 #define fdimf32 flt32_fdim
790 #define fdimf64x flt64x_fdim
791 #define fdimf64 flt64_fdim
792 #define fdivl ldbl_toflt_div
793 #define fdiv toflt_div
794 #define fegetmode fe_getmode
795 #define fesetexcept fe_setexcept
796 #define fesetmode fe_setmode
797 #define fetestexceptflag fe_testexceptflag
798 #define ffma1 ldbl_toflt_fma
799 #define ffma toflt_fma
800 #define floord128x dec128x_floor
801 #define floord128 dec128_floor
802 #define floord32 dec32_floor
803 #define floorf128x flt128x_floor
804 #define floorf16 flt16_floor
805 #define floorf32x flt32x_floor
806 #define floorf32 flt32_floor
807 #define fmad128x dec128x_fma
808 #define fmad128 dec128_fma
809 #define fmad32 dec32_fma
810 #define fmad64x dec64x_fma
811 #define fmad64 dec64_fma
812 #define fmaf128x flt128x_fma
813 #define fmaf128 flt128_fma
814 #define fmaf16 flt16_fma
815 #define fmaf32x flt32x_fma
816 #define fmaf32 flt32_fma
817 #define fmaf64x flt64x_fma
```

```
818 #define fmaf64 flt64_fma
819 #define fmaxd128x dec128x_fmax
820 #define fmaxd128 dec128_fmax
821 #define fmaxd32 dec32_fmax
822 #define fmaxd64x dec64x_fmax
823 #define fmaxd64 dec64_fmax
824 #define fmaxf128x flt128x_fmax
825 #define fmaxf128 flt128_fmax
826 #define fmaxf16 flt16_fmax
827 #define fmaxf32x flt32x_fmax
828 #define fmaxf32 flt32_fmax
829 #define fmaxf64x flt64x_fmax
830 #define fmaxf64 flt64_fmax
831 #define fmaxmagd128x dec128x_fmaxmag
832 #define fmaxmagd128 dec128_fmaxmag
833 #define fmaxmagd32 dec32_fmaxmag
834 #define fmaxmagd64x dec64x_fmaxmag
835 #define fmaxmagd64 dec64_fmaxmag
836 #define fmaxmagf128x flt128x_fmaxmag
837 #define fmaxmagf128 flt128_fmaxmag
838 #define fmaxmagf16 flt16_fmaxmag
839 #define fmaxmagf32x flt32x_fmaxmag
840 #define fmaxmagf32 flt32_fmaxmag
841 #define fmaxmagf64x flt64x_fmaxmag
842 #define fmaxmagf64 flt64_fmaxmag
843 #define fmaxmagf flt_fmaxmag
844 #define fmaxmagl ldbl_fmaxmag
845 #define fmind128x dec128x_fmin
846 #define fmind128 dec128_fmin
847 #define fmind32 dec32_fmin
848 #define fmind64x dec64x_fmin
849 #define fmind64 dec64_fmin
850 #define fminf128x flt128x_fmin
851 #define fminf128 flt128_fmin
852 #define fminf16 flt16_fmin
853 #define fminf32x flt32x_fmin
854 #define fminf32 flt32_fmin
855 #define fminf64x flt64x_fmin
856 #define fminf64 flt64_fmin
857 #define fminmagd128x dec128x_fminmag
858 #define fminmagd128 dec128_fminmag
859 #define fminmagd32 dec32_fminmag
860 #define fminmagd64x dec64x_fminmag
861 #define fminmagd64 dec64_fminmag
862 #define fminmagf128x flt128x_fminmag
863 #define fminmagf128 flt128_fminmag
864 #define fminmagf16 flt16_fminmag
865 #define fminmagf32x flt32x_fminmag
866 #define fminmagf32 flt32_fminmag
867 #define fminmagf64x flt64x_fminmag
868 #define fminmagf64 flt64_fminmag
869 #define fminmagf flt_fminmag
870 #define fminmagl ldbl_fminmag
871 #define fmodd128x dec128x_fmod
872 #define fmodd128 dec128_fmod
873 #define fmodd32 dec32_fmod
874 #define fmodd64x dec64x_fmod
875 #define fmodd64 dec64_fmod
876 #define fmodf128x flt128x_fmod
877 #define fmodf128 flt128_fmod
878 #define fmodf16 flt16_fmod
879 #define fmodf32x flt32x_fmod
880 #define fmodf32 flt32_fmod
```

```
881 #define fmodf64x flt64x_fmod
882 #define fmodf64 flt64_fmod
883 #define fmul1 ldbl_toflt_mul
884 #define fmul toflt_mul
885 #define frexpd128x dec128x_frexp
886 #define frexpd128 dec128_frexp
887 #define frexpd32 dec32_frexp
888 #define frexpd64x dec64x_frexp
889 #define frexpd64 dec64_frexp
890 #define frexpf128x flt128x_frexp
891 #define frexpf128 flt128_frexp
892 #define frexpf16 flt16_frexp
893 #define frexpf32x flt32x_frexp
894 #define frexpf32 flt32_frexp
895 #define frexpf64x flt64x_frexp
896 #define frexpf64 flt64_frexp
897 #define fromfpd128x dec128x_toint
898 #define fromfpd128 dec128_toint
899 #define fromfpd32 dec32_toint
900 #define fromfpd64x dec64x_toint
901 #define fromfpd64 dec64_toint
902 #define fromfpf128x flt128x_toint
903 #define fromfpf128 flt128_toint
904 #define fromfpf16 flt16_toint
905 #define fromfpf32x flt32x_toint
906 #define fromfpf32 flt32_toint
907 #define fromfpf64x flt64x_toint
908 #define fromfpf64 flt64_toint
909 #define fromfpf flt_toint
910 #define fromfpl ldbl_toint
911 #define fromfpdx128x dec128x_tointx
912 #define fromfpdx128 dec128_tointx
913 #define fromfpdx32 dec32_tointx
914 #define fromfpdx64x dec64x_tointx
915 #define fromfpdx64 dec64_tointx
916 #define fromfpxf128x flt128x_tointx
917 #define fromfpxf128 flt128_tointx
918 #define fromfpxf16 flt16_tointx
919 #define fromfpxf32x flt32x_tointx
920 #define fromfpxf32 flt32_tointx
921 #define fromfpxf64x flt64x_tointx
922 #define fromfpxf64 flt64_tointx
923 #define fromfpxf flt_tointx
924 #define fromfpx1 ldbl_tointx
925 #define fromfpx tointx
926 #define fromfp toint
927 #define fsqrt1 ldbl_toflt_sqrt
928 #define fsqrt toflt_sqrt
929 #define fsub1 ldbl_toflt_sub
930 #define fsub toflt_sub
931 #define getpayloadd128x dec128x_getpayload
932 #define getpayloadd128 dec128_getpayload
933 #define getpayloadd32 dec32_getpayload
934 #define getpayloadd64x dec64x_getpayload
935 #define getpayloadd64 dec64_getpayload
936 #define getpayloadf128x flt128x_getpayload
937 #define getpayloadf128 flt128_getpayload
938 #define getpayloadf16 flt16_getpayload
939 #define getpayloadf32x flt32x_getpayload
940 #define getpayloadf32 flt32_getpayload
941 #define getpayloadf64x flt64x_getpayload
942 #define getpayloadf64 flt64_getpayload
943 #define getpayloadf flt_getpayload
```

```
944 #define getpayload1 ldbl_getpayload
945 #define hypotd128x dec128x_hypot
946 #define hypotd128 dec128_hypot
947 #define hypotd32 dec32_hypot
948 #define hypotd64x dec64x_hypot
949 #define hypotd64 dec64_hypot
950 #define hypotf128x flt128x_hypot
951 #define hypotf128 flt128_hypot
952 #define hypotf16 flt16_hypot
953 #define hypotf32x flt32x_hypot
954 #define hypotf32 flt32_hypot
955 #define hypotf64x flt64x_hypot
956 #define hypotf64 flt64_hypot
957 #define ilogbd128x dec128x_ilogb
958 #define ilogbd128 dec128_ilogb
959 #define ilogbd32 dec32_ilogb
960 #define ilogbd64x dec64x_ilogb
961 #define ilogbd64 dec64_ilogb
962 #define ilogbf128x flt128x_ilogb
963 #define ilogbf128 flt128_ilogb
964 #define ilogbf16 flt16_ilogb
965 #define ilogbf32x flt32x_ilogb
966 #define ilogbf32 flt32_ilogb
967 #define ilogbf64x flt64x_ilogb
968 #define ilogbf64 flt64_ilogb
969 #define ldexpd128x dec128x_ldexp
970 #define ldexpd128 dec128_ldexp
971 #define ldexpd32 dec32_ldexp
972 #define ldexpd64x dec64x_ldexp
973 #define ldexpd64 dec64_ldexp
974 #define ldexpf128x flt128x_ldexp
975 #define ldexpf128 flt128_ldexp
976 #define ldexpf16 flt16_ldexp
977 #define ldexpf32x flt32x_ldexp
978 #define ldexpf32 flt32_ldexp
979 #define ldexpf64x flt64x_ldexp
980 #define ldexpf64 flt64_ldexp
981 #define lgammad128x dec128x_lgamma
982 #define lgammad128 dec128_lgamma
983 #define lgammad32 dec32_lgamma
984 #define lgammad64x dec64x_lgamma
985 #define lgammad64 dec64_lgamma
986 #define lgammaf128x flt128x_lgamma
987 #define lgammaf128 flt128_lgamma
988 #define lgammaf16 flt16_lgamma
989 #define lgammaf32x flt32x_lgamma
990 #define lgammaf32 flt32_lgamma
991 #define lgammaf64x flt64x_lgamma
992 #define lgammaf64 flt64_lgamma
993 #define llogbd128x dec128x_toint_logb
994 #define llogbd128 dec128_toint_logb
995 #define llogbd32 dec32_toint_logb
996 #define llogbd64x dec64x_toint_logb
997 #define llogbd64 dec64_toint_logb
998 #define llogbf128x flt128x_toint_logb
999 #define llogbf128 flt128_toint_logb
1000 #define llogbf16 flt16_toint_logb
1001 #define llogbf32x flt32x_toint_logb
1002 #define llogbf32 flt32_toint_logb
1003 #define llogbf64x flt64x_toint_logb
1004 #define llogbf64 flt64_toint_logb
1005 #define llogbf flt_toint_logb
1006 #define llogbl ldbl_toint_logb
```

```
1007 #define llogb toint_logb
1008 #define llquantexpd128x dec128x_toint_quantexp
1009 #define llquantexpd128 dec128_toint_quantexp
1010 #define llquantexpd32 dec32_toint_quantexp
1011 #define llquantexpd64x dec64x_toint_quantexp
1012 #define llquantexpd64 dec64_toint_quantexp
1013 #define llquantexpf128x flt128x_toint_quantexp
1014 #define llquantexpf128 flt128_toint_quantexp
1015 #define llquantexpf16 flt16_toint_quantexp
1016 #define llquantexpf32x flt32x_toint_quantexp
1017 #define llquantexpf32 flt32_toint_quantexp
1018 #define llquantexpf64x flt64x_toint_quantexp
1019 #define llquantexpf64 flt64_toint_quantexp
1020 #define llquantexp toint_quantexp
1021 #define llrintd128x dec128x_llrint
1022 #define llrintd128 dec128_llrint
1023 #define llrintd32 dec32_llrint
1024 #define llrintd64x dec64x_llrint
1025 #define llrintd64 dec64_llrint
1026 #define llrintf128x flt128x_llrint
1027 #define llrintf128 flt128_llrint
1028 #define llrintf16 flt16_llrint
1029 #define llrintf32x flt32x_llrint
1030 #define llrintf32 flt32_llrint
1031 #define llrintf64x flt64x_llrint
1032 #define llrintf64 flt64_llrint
1033 #define llroundd128x dec128x_llround
1034 #define llroundd128 dec128_llround
1035 #define llroundd32 dec32_llround
1036 #define llroundd64x dec64x_llround
1037 #define llroundd64 dec64_llround
1038 #define llroundf128x flt128x_llround
1039 #define llroundf128 flt128_llround
1040 #define llroundf16 flt16_llround
1041 #define llroundf32x flt32x_llround
1042 #define llroundf32 flt32_llround
1043 #define llroundf64x flt64x_llround
1044 #define llroundf64 flt64_llround
1045 #define log10d128x dec128x_log10
1046 #define log10d128 dec128_log10
1047 #define log10d32 dec32_log10
1048 #define log10d64x dec64x_log10
1049 #define log10d64 dec64_log10
1050 #define log10f128x flt128x_log10
1051 #define log10f128 flt128_log10
1052 #define log10f32x flt32x_log10
1053 #define log10f32 flt32_log10
1054 #define log10f64x flt64x_log10
1055 #define log10f64 flt64_log10
1056 #define log10pd128x dec128x_log10p1
1057 #define log10pd128 dec128_log10p1
1058 #define log10pd32 dec32_log10p1
1059 #define log10pd64x dec64x_log10p1
1060 #define log10pd64 dec64_log10p1
1061 #define log10p1f128x flt128x_log10p1
1062 #define log10p1f128 flt128_log10p1
1063 #define log10p1f32x flt32x_log10p1
1064 #define log10p1f32 flt32_log10p1
1065 #define log10p1f64x flt64x_log10p1
1066 #define log10p1f64 flt64_log10p1
1067 #define log10p1f flt_log10p1
1068 #define log10p11 ldbl_log10p1
1069 #define log1pd128x dec128x_log1p
```

```
1070 #define log1pd128 dec128_log1p
1071 #define log1pd32 dec32_log1p
1072 #define log1pd64x dec64x_log1p
1073 #define log1pd64 dec64_log1p
1074 #define log1pf128x flt128x_log1p
1075 #define log1pf128 flt128_log1p
1076 #define log1pf32x flt32x_log1p
1077 #define log1pf32 flt32_log1p
1078 #define log1pf64x flt64x_log1p
1079 #define log1pf64 flt64_log1p
1080 #define log2d128x dec128x_log2
1081 #define log2d128 dec128_log2
1082 #define log2d32 dec32_log2
1083 #define log2d64x dec64x_log2
1084 #define log2d64 dec64_log2
1085 #define log2f128x flt128x_log2
1086 #define log2f128 flt128_log2
1087 #define log2f32x flt32x_log2
1088 #define log2f32 flt32_log2
1089 #define log2f64x flt64x_log2
1090 #define log2f64 flt64_log2
1091 #define log2p1d128x dec128x_log2p1
1092 #define log2p1d128 dec128_log2p1
1093 #define log2p1d32 dec32_log2p1
1094 #define log2p1d64x dec64x_log2p1
1095 #define log2p1d64 dec64_log2p1
1096 #define log2p1f128x flt128x_log2p1
1097 #define log2p1f128 flt128_log2p1
1098 #define log2p1f32x flt32x_log2p1
1099 #define log2p1f32 flt32_log2p1
1100 #define log2p1f64x flt64x_log2p1
1101 #define log2p1f64 flt64_log2p1
1102 #define log2p1f flt_log2p1
1103 #define log2p1 ldbl_log2p1
1104 #define logbd128x dec128x_logb
1105 #define logbd128 dec128_logb
1106 #define logbd32 dec32_logb
1107 #define logbd64x dec64x_logb
1108 #define logbd64 dec64_logb
1109 #define logbf128x flt128x_logb
1110 #define logbf128 flt128_logb
1111 #define logbf16 flt16_logb
1112 #define logbf32x flt32x_logb
1113 #define logbf32 flt32_logb
1114 #define logbf64x flt64x_logb
1115 #define logbf64 flt64_logb
1116 #define logd128x dec128x_log
1117 #define logd128 dec128_log
1118 #define logd32 dec32_log
1119 #define logd64x dec64x_log
1120 #define logd64 dec64_log
1121 #define logf128x flt128x_log
1122 #define logf128 flt128_log
1123 #define logf16 flt16_log
1124 #define logf32x flt32x_log
1125 #define logf32 flt32_log
1126 #define logf64x flt64x_log
1127 #define logf64 flt64_log
1128 #define logp1d128x dec128x_logp1
1129 #define logp1d128 dec128_logp1
1130 #define logp1d32 dec32_logp1
1131 #define logp1d64x dec64x_logp1
1132 #define logp1d64 dec64_logp1
```

```
1133 #define logp1f128x flt128x_logp1
1134 #define logp1f128 flt128_logp1
1135 #define logp1f32x flt32x_logp1
1136 #define logp1f32 flt32_logp1
1137 #define logp1f64x flt64x_logp1
1138 #define logp1f64 flt64_logp1
1139 #define logp1f flt_logp1
1140 #define logp1l ldbl_logp1
1141 #define lrintd128x dec128x_lrint
1142 #define lrintd128 dec128_lrint
1143 #define lrintd32 dec32_lrint
1144 #define lrintd64x dec64x_lrint
1145 #define lrintd64 dec64_lrint
1146 #define lrintf128x flt128x_lrint
1147 #define lrintf128 flt128_lrint
1148 #define lrintf16 flt16_lrint
1149 #define lrintf32x flt32x_lrint
1150 #define lrintf32 flt32_lrint
1151 #define lrintf64x flt64x_lrint
1152 #define lrintf64 flt64_lrint
1153 #define lroundd128x dec128x_lround
1154 #define lroundd128 dec128_lround
1155 #define lroundd32 dec32_lround
1156 #define lroundd64x dec64x_lround
1157 #define lroundd64 dec64_lround
1158 #define lroundf128x flt128x_lround
1159 #define lroundf128 flt128_lround
1160 #define lroundf16 flt16_lround
1161 #define lroundf32x flt32x_lround
1162 #define lroundf32 flt32_lround
1163 #define lroundf64x flt64x_lround
1164 #define lroundf64 flt64_lround
1165 #define modfd128x dec128x_modf
1166 #define modfd128 dec128_modf
1167 #define modfd32 dec32_modf
1168 #define modfd64x dec64x_modf
1169 #define modfd64 dec64_modf
1170 #define modff128x flt128x_modf
1171 #define modff128 flt128_modf
1172 #define modff16 flt16_modf
1173 #define modff32x flt32x_modf
1174 #define modff32 flt32_modf
1175 #define modff64x flt64x_modf
1176 #define modff64 flt64_modf
1177 #define nand128x dec128x_nan
1178 #define nand128 dec128_nan
1179 #define nand32 dec32_nan
1180 #define nand64x dec64x_nan
1181 #define nand64 dec64_nan
1182 #define nanf128x flt128x_nan
1183 #define nanf128 flt128_nan
1184 #define nanf16 flt16_nan
1185 #define nanf32x flt32x_nan
1186 #define nanf32 flt32_nan
1187 #define nanf64x flt64x_nan
1188 #define nanf64 flt64_nan
1189 #define nearbyintd128x dec128x_nearbyint
1190 #define nearbyintd128 dec128_nearbyint
1191 #define nearbyintd32 dec32_nearbyint
1192 #define nearbyintd64x dec64x_nearbyint
1193 #define nearbyintd64 dec64_nearbyint
1194 #define nearbyintf128x flt128x_nearbyint
1195 #define nearbyintf128 flt128_nearbyint
```



```
1196 #define nearbyintf16 flt16_nearbyint
1197 #define nearbyintf32x flt32x_nearbyint
1198 #define nearbyintf32 flt32_nearbyint
1199 #define nearbyintf64x flt64x_nearbyint
1200 #define nearbyintf64 flt64_nearbyint
1201 #define nextafterd128x dec128x_nextafter
1202 #define nextafterd128 dec128_nextafter
1203 #define nextafterd32 dec32_nextafter
1204 #define nextafterd64x dec64x_nextafter
1205 #define nextafterd64 dec64_nextafter
1206 #define nextafterf128x flt128x_nextafter
1207 #define nextafterf128 flt128_nextafter
1208 #define nextafterf16 flt16_nextafter
1209 #define nextafterf32x flt32x_nextafter
1210 #define nextafterf32 flt32_nextafter
1211 #define nextafterf64x flt64x_nextafter
1212 #define nextafterf64 flt64_nextafter
1213 #define nextdownnd128x dec128x_nextdown
1214 #define nextdownnd128 dec128_nextdown
1215 #define nextdownnd32 dec32_nextdown
1216 #define nextdownnd64x dec64x_nextdown
1217 #define nextdownnd64 dec64_nextdown
1218 #define nextdownnf128x flt128x_nextdown
1219 #define nextdownnf128 flt128_nextdown
1220 #define nextdownnf16 flt16_nextdown
1221 #define nextdownnf32x flt32x_nextdown
1222 #define nextdownnf32 flt32_nextdown
1223 #define nextdownnf64x flt64x_nextdown
1224 #define nextdownnf64 flt64_nextdown
1225 #define nextdownnf flt_nextdown
1226 #define nextdownnl ldbl_nextdown
1227 #define nexttowardd128x dec128x_nexttoward
1228 #define nexttowardd128 dec128_nexttoward
1229 #define nexttowardd32 dec32_nexttoward
1230 #define nexttowardd64x dec64x_nexttoward
1231 #define nexttowardd64 dec64_nexttoward
1232 #define nexttowardf128x flt128x_nexttoward
1233 #define nexttowardf128 flt128_nexttoward
1234 #define nexttowardf16 flt16_nexttoward
1235 #define nexttowardf32x flt32x_nexttoward
1236 #define nexttowardf32 flt32_nexttoward
1237 #define nexttowardf64x flt64x_nexttoward
1238 #define nexttowardf64 flt64_nexttoward
1239 #define nextupd128x dec128x_nextup
1240 #define nextupd128 dec128_nextup
1241 #define nextupd32 dec32_nextup
1242 #define nextupd64x dec64x_nextup
1243 #define nextupd64 dec64_nextup
1244 #define nextupf128x flt128x_nextup
1245 #define nextupf128 flt128_nextup
1246 #define nextupf16 flt16_nextup
1247 #define nextupf32x flt32x_nextup
1248 #define nextupf32 flt32_nextup
1249 #define nextupf64x flt64x_nextup
1250 #define nextupf64 flt64_nextup
1251 #define nextupf flt_nextup
1252 #define nextupl ldbl_nextup
1253 #define powd128x dec128x_pow
1254 #define powd128 dec128_pow
1255 #define powd32 dec32_pow
1256 #define powd64x dec64x_pow
1257 #define powd64 dec64_pow
1258 #define powf128x flt128x_pow
```

```
1259 #define powf128 flt128_pow
1260 #define powf16 flt16_pow
1261 #define powf32x flt32x_pow
1262 #define powf32 flt32_pow
1263 #define powf64x flt64x_pow
1264 #define powf64 flt64_pow
1265 #define pownd128x dec128x_pown
1266 #define pownd128 dec128_pown
1267 #define pownd32 dec32_pown
1268 #define pownd64x dec64x_pown
1269 #define pownd64 dec64_pown
1270 #define pownf128x flt128x_pown
1271 #define pownf128 flt128_pown
1272 #define pownf16 flt16_pown
1273 #define pownf32x flt32x_pown
1274 #define pownf32 flt32_pown
1275 #define pownf64x flt64x_pown
1276 #define pownf64 flt64_pown
1277 #define pownf flt_pown
1278 #define pownl ldbl_pown
1279 #define powrd128x dec128x_powr
1280 #define powrd128 dec128_powr
1281 #define powrd32 dec32_powr
1282 #define powrd64x dec64x_powr
1283 #define powrd64 dec64_powr
1284 #define powrf128x flt128x_powr
1285 #define powrf128 flt128_powr
1286 #define powrf16 flt16_powr
1287 #define powrf32x flt32x_powr
1288 #define powrf32 flt32_powr
1289 #define powrf64x flt64x_powr
1290 #define powrf64 flt64_powr
1291 #define powrf flt_powr
1292 #define powrl ldbl_powr
1293 #define quantized128x dec128x_quantize
1294 #define quantized128 dec128_quantize
1295 #define quantized32 dec32_quantize
1296 #define quantized64x dec64x_quantize
1297 #define quantized64 dec64_quantize
1298 #define quantizef128x flt128x_quantize
1299 #define quantizef128 flt128_quantize
1300 #define quantizef16 flt16_quantize
1301 #define quantizef32x flt32x_quantize
1302 #define quantizef32 flt32_quantize
1303 #define quantizef64x flt64x_quantize
1304 #define quantizef64 flt64_quantize
1305 #define quantumd128x dec128x_quantum
1306 #define quantumd128 dec128_quantum
1307 #define quantumd32 dec32_quantum
1308 #define quantumd64x dec64x_quantum
1309 #define quantumd64 dec64_quantum
1310 #define quantumf128x flt128x_quantum
1311 #define quantumf128 flt128_quantum
1312 #define quantumf16 flt16_quantum
1313 #define quantumf32x flt32x_quantum
1314 #define quantumf32 flt32_quantum
1315 #define quantumf64x flt64x_quantum
1316 #define quantumf64 flt64_quantum
1317 #define remainderd128x dec128x_remainder
1318 #define remainderd128 dec128_remainder
1319 #define remainderd32 dec32_remainder
1320 #define remainderd64x dec64x_remainder
1321 #define remainderd64 dec64_remainder
```

```
1322 #define remainderf128x flt128x_remainder
1323 #define remainderf128 flt128_remainder
1324 #define remainderf16 flt16_remainder
1325 #define remainderf32x flt32x_remainder
1326 #define remainderf32 flt32_remainder
1327 #define remainderf64x flt64x_remainder
1328 #define remainderf64 flt64_remainder
1329 #define rintd128x dec128x_rint
1330 #define rintd128 dec128_rint
1331 #define rintd32 dec32_rint
1332 #define rintd64x dec64x_rint
1333 #define rintd64 dec64_rint
1334 #define rintf128x flt128x_rint
1335 #define rintf128 flt128_rint
1336 #define rintf16 flt16_rint
1337 #define rintf32x flt32x_rint
1338 #define rintf32 flt32_rint
1339 #define rintf64x flt64x_rint
1340 #define rintf64 flt64_rint
1341 #define rootnd128x dec128x_rootn
1342 #define rootnd128 dec128_rootn
1343 #define rootnd32 dec32_rootn
1344 #define rootnd64x dec64x_rootn
1345 #define rootnd64 dec64_rootn
1346 #define rootnf128x flt128x_rootn
1347 #define rootnf128 flt128_rootn
1348 #define rootnf16 flt16_rootn
1349 #define rootnf32x flt32x_rootn
1350 #define rootnf32 flt32_rootn
1351 #define rootnf64x flt64x_rootn
1352 #define rootnf64 flt64_rootn
1353 #define rootnf flt_rootn
1354 #define rootnl ldbl_rootn
1355 #define roundd128x dec128x_round
1356 #define roundd128 dec128_round
1357 #define roundd32 dec32_round
1358 #define roundd64x dec64x_round
1359 #define roundd64 dec64_round
1360 #define roundevend128x dec128x_roundeven
1361 #define roundevend128 dec128_roundeven
1362 #define roundevend32 dec32_roundeven
1363 #define roundevend64x dec64x_roundeven
1364 #define roundevend64 dec64_roundeven
1365 #define roundevenf128x flt128x_roundeven
1366 #define roundevenf128 flt128_roundeven
1367 #define roundevenf16 flt16_roundeven
1368 #define roundevenf32x flt32x_roundeven
1369 #define roundevenf32 flt32_roundeven
1370 #define roundevenf64x flt64x_roundeven
1371 #define roundevenf64 flt64_roundeven
1372 #define roundevenf flt_roundeven
1373 #define roundevenl ldbl_roundeven
1374 #define roundf128x flt128x_round
1375 #define roundf128 flt128_round
1376 #define roundf16 flt16_round
1377 #define roundf32x flt32x_round
1378 #define roundf32 flt32_round
1379 #define roundf64x flt64x_round
1380 #define roundf64 flt64_round
1381 #define rsqrd128x dec128x_rsqrtd
1382 #define rsqrd128 dec128_rsqrtd
1383 #define rsqrd32 dec32_rsqrtd
1384 #define rsqrd64x dec64x_rsqrtd
```

```
1385 #define rsqrtd64 dec64_rsqrtd
1386 #define rsqrtdf128x flt128x_rsqrtd
1387 #define rsqrtdf128 flt128_rsqrtd
1388 #define rsqrtdf16 flt16_rsqrtd
1389 #define rsqrtdf32x flt32x_rsqrtd
1390 #define rsqrtdf32 flt32_rsqrtd
1391 #define rsqrtdf64x flt64x_rsqrtd
1392 #define rsqrtdf64 flt64_rsqrtd
1393 #define rsqrtdf flt_rsqrtd
1394 #define rsqrtdl ldbl_rsqrtd
1395 #define samequantumd128x dec128x_samequantumd
1396 #define samequantumd128 dec128_samequantumd
1397 #define samequantumd32 dec32_samequantumd
1398 #define samequantumd64x dec64x_samequantumd
1399 #define samequantumd64 dec64_samequantumd
1400 #define samequantumf128x flt128x_samequantumf
1401 #define samequantumf128 flt128_samequantumf
1402 #define samequantumf16 flt16_samequantumf
1403 #define samequantumf32x flt32x_samequantumf
1404 #define samequantumf32 flt32_samequantumf
1405 #define samequantumf64x flt64x_samequantumf
1406 #define samequantumf64 flt64_samequantumf
1407 #define scalblnd128x dec128x_scalblnd
1408 #define scalblnd128 dec128_scalblnd
1409 #define scalblnd32 dec32_scalblnd
1410 #define scalblnd64x dec64x_scalblnd
1411 #define scalblnd64 dec64_scalblnd
1412 #define scalblnf128x flt128x_scalblnf
1413 #define scalblnf128 flt128_scalblnf
1414 #define scalblnf16 flt16_scalblnf
1415 #define scalblnf32x flt32x_scalblnf
1416 #define scalblnf32 flt32_scalblnf
1417 #define scalblnf64x flt64x_scalblnf
1418 #define scalblnf64 flt64_scalblnf
1419 #define scalbnd128x dec128x_scalbnd
1420 #define scalbnd128 dec128_scalbnd
1421 #define scalbnd32 dec32_scalbnd
1422 #define scalbnd64x dec64x_scalbnd
1423 #define scalbnd64 dec64_scalbnd
1424 #define scalbnf128x flt128x_scalbnf
1425 #define scalbnf128 flt128_scalbnf
1426 #define scalbnf16 flt16_scalbnf
1427 #define scalbnf32x flt32x_scalbnf
1428 #define scalbnf32 flt32_scalbnf
1429 #define scalbnf64x flt64x_scalbnf
1430 #define scalbnf64 flt64_scalbnf
1431 #define setpayloadd128x dec128x_setpayloadd
1432 #define setpayloadd128 dec128_setpayloadd
1433 #define setpayloadd32 dec32_setpayloadd
1434 #define setpayloadd64x dec64x_setpayloadd
1435 #define setpayloadd64 dec64_setpayloadd
1436 #define setpayloadf128x flt128x_setpayloadf
1437 #define setpayloadf128 flt128_setpayloadf
1438 #define setpayloadf16 flt16_setpayloadf
1439 #define setpayloadf32x flt32x_setpayloadf
1440 #define setpayloadf32 flt32_setpayloadf
1441 #define setpayloadf64x flt64x_setpayloadf
1442 #define setpayloadf64 flt64_setpayloadf
1443 #define setpayloadf flt_setpayloadf
1444 #define setpayloadl ldbl_setpayloadl
1445 #define setpayloadsigd128x dec128x_setpayloadsigd
1446 #define setpayloadsigd128 dec128_setpayloadsigd
1447 #define setpayloadsigd32 dec32_setpayloadsigd
```

```
1448 #define setpayloadsigd64x dec64x_setpayloadsig
1449 #define setpayloadsigd64 dec64_setpayloadsig
1450 #define setpayloadsigf128x flt128x_setpayloadsig
1451 #define setpayloadsigf128 flt128_setpayloadsig
1452 #define setpayloadsigf16 flt16_setpayloadsig
1453 #define setpayloadsigf32x flt32x_setpayloadsig
1454 #define setpayloadsigf32 flt32_setpayloadsig
1455 #define setpayloadsigf64x flt64x_setpayloadsig
1456 #define setpayloadsigf64 flt64_setpayloadsig
1457 #define setpayloadsigf flt_setpayloadsig
1458 #define setpayloadsigl ldbl_setpayloadsig
1459 #define sind128x dec128x_sin
1460 #define sind128 dec128_sin
1461 #define sind32 dec32_sin
1462 #define sind64x dec64x_sin
1463 #define sind64 dec64_sin
1464 #define sinf128x flt128x_sin
1465 #define sinf128 flt128_sin
1466 #define sinf16 flt16_sin
1467 #define sinf32x flt32x_sin
1468 #define sinf32 flt32_sin
1469 #define sinf64x flt64x_sin
1470 #define sinf64 flt64_sin
1471 #define sinh128x dec128x_sinh
1472 #define sinh128 dec128_sinh
1473 #define sinh32 dec32_sinh
1474 #define sinh64x dec64x_sinh
1475 #define sinh64 dec64_sinh
1476 #define sinhf128x flt128x_sinh
1477 #define sinhf128 flt128_sinh
1478 #define sinhf16 flt16_sinh
1479 #define sinhf32x flt32x_sinh
1480 #define sinhf32 flt32_sinh
1481 #define sinhf64x flt64x_sinh
1482 #define sinhf64 flt64_sinh
1483 #define sinpid128x dec128x_sinpi
1484 #define sinpid128 dec128_sinpi
1485 #define sinpid32 dec32_sinpi
1486 #define sinpid64x dec64x_sinpi
1487 #define sinpid64 dec64_sinpi
1488 #define sinpif128x flt128x_sinpi
1489 #define sinpif128 flt128_sinpi
1490 #define sinpif16 flt16_sinpi
1491 #define sinpif32x flt32x_sinpi
1492 #define sinpif32 flt32_sinpi
1493 #define sinpif64x flt64x_sinpi
1494 #define sinpif64 flt64_sinpi
1495 #define sinpif flt_sinpi
1496 #define sinpil ldbl_sinpi
1497 #define sqrtd128x dec128x_sqrt
1498 #define sqrtd128 dec128_sqrt
1499 #define sqrtd32 dec32_sqrt
1500 #define sqrtd64x dec64x_sqrt
1501 #define sqrtd64 dec64_sqrt
1502 #define sqrtf128x flt128x_sqrt
1503 #define sqrtf128 flt128_sqrt
1504 #define sqrtf16 flt16_sqrt
1505 #define sqrtf32x flt32x_sqrt
1506 #define sqrtf32 flt32_sqrt
1507 #define sqrtf64x flt64x_sqrt
1508 #define sqrtf64 flt64_sqrt
1509 #define tand128x dec128x_tan
1510 #define tand128 dec128_tan
```

```
1511 #define tand32 dec32_tan
1512 #define tand64x dec64x_tan
1513 #define tand64 dec64_tan
1514 #define tanf128x flt128x_tan
1515 #define tanf128 flt128_tan
1516 #define tanf16 flt16_tan
1517 #define tanf32x flt32x_tan
1518 #define tanf32 flt32_tan
1519 #define tanf64x flt64x_tan
1520 #define tanf64 flt64_tan
1521 #define tanhd128x dec128x_tanh
1522 #define tanhd128 dec128_tanh
1523 #define tanhd32 dec32_tanh
1524 #define tanhd64x dec64x_tanh
1525 #define tanhd64 dec64_tanh
1526 #define tanhf128x flt128x_tanh
1527 #define tanhf128 flt128_tanh
1528 #define tanhf16 flt16_tanh
1529 #define tanhf32x flt32x_tanh
1530 #define tanhf32 flt32_tanh
1531 #define tanhf64x flt64x_tanh
1532 #define tanhf64 flt64_tanh
1533 #define tanpid128x dec128x_tanpi
1534 #define tanpid128 dec128_tanpi
1535 #define tanpid32 dec32_tanpi
1536 #define tanpid64x dec64x_tanpi
1537 #define tanpid64 dec64_tanpi
1538 #define tanpif128x flt128x_tanpi
1539 #define tanpif128 flt128_tanpi
1540 #define tanpif16 flt16_tanpi
1541 #define tanpif32x flt32x_tanpi
1542 #define tanpif32 flt32_tanpi
1543 #define tanpif64x flt64x_tanpi
1544 #define tanpif64 flt64_tanpi
1545 #define tanpif flt_tanpi
1546 #define tanpil ldbl_tanpi
1547 #define tgamma128x dec128x_tgamma
1548 #define tgamma128 dec128_tgamma
1549 #define tgamma32 dec32_tgamma
1550 #define tgamma64x dec64x_tgamma
1551 #define tgamma64 dec64_tgamma
1552 #define tgammaf128x flt128x_tgamma
1553 #define tgammaf128 flt128_tgamma
1554 #define tgammaf16 flt16_tgamma
1555 #define tgammaf32x flt32x_tgamma
1556 #define tgammaf32 flt32_tgamma
1557 #define tgammaf64x flt64x_tgamma
1558 #define tgammaf64 flt64_tgamma
1559 #define truncd128x dec128x_trunc
1560 #define truncd128 dec128_trunc
1561 #define truncd32 dec32_trunc
1562 #define truncd64x dec64x_trunc
1563 #define truncd64 dec64_trunc
1564 #define truncf128x flt128x_trunc
1565 #define truncf128 flt128_trunc
1566 #define truncf16 flt16_trunc
1567 #define truncf32x flt32x_trunc
1568 #define truncf32 flt32_trunc
1569 #define truncf64x flt64x_trunc
1570 #define truncf64 flt64_trunc
1571 #define ufromfpd128x dec128x_toint
1572 #define ufromfpd128 dec128_toint
1573 #define ufromfpd32 dec32_toint
```

```
1574 #define ufromfpd64x dec64x_touint
1575 #define ufromfpd64 dec64_touint
1576 #define ufromfpf128x flt128x_touint
1577 #define ufromfpf128 flt128_touint
1578 #define ufromfpf16 flt16_touint
1579 #define ufromfpf32x flt32x_touint
1580 #define ufromfpf32 flt32_touint
1581 #define ufromfpf64x flt64x_touint
1582 #define ufromfpf64 flt64_touint
1583 #define ufromfpf flt_touint
1584 #define ufromfpl ldbl_touint
1585 #define ufromfpxd128x dec128x_touintx
1586 #define ufromfpxd128 dec128_touintx
1587 #define ufromfpxd32 dec32_touintx
1588 #define ufromfpxd64x dec64x_touintx
1589 #define ufromfpxd64 dec64_touintx
1590 #define ufromfpxf128x flt128x_touintx
1591 #define ufromfpxf128 flt128_touintx
1592 #define ufromfpxf16 flt16_touintx
1593 #define ufromfpxf32x flt32x_touintx
1594 #define ufromfpxf32 flt32_touintx
1595 #define ufromfpxf64x flt64x_touintx
1596 #define ufromfpxf64 flt64_touintx
1597 #define ufromfpxf flt_touintx
1598 #define ufromfpxl ldbl_touintx
1599 #define ufromfpx touintx
1600 #define ufromfp touint
1601 #define _Decimal132_t dec32_eval_t
1602 #define _Decimal64_t dec64_eval_t
1603 #define _Decimal128_t dec128_eval_t
1604 #define _Float32_t flt32_eval_t
1605 #define _Float64_t flt64_eval_t
1606 #define _Float128_t flt128_eval_t
1607 #define _Decimal132 dec32_t
1608 #define _Decimal64 dec64_t
1609 #define _Decimal128 dec128_t
1610 #define _Float32 flt32_t
1611 #define _Float64 flt64_t
1612 #define _Float128 flt128_t
1613 #define _Decimal32x dec32x_t
1614 #define _Decimal64x dec64x_t
1615 #define _Decimal128x dec128x_t
1616 #define _Float32x flt32x_t
1617 #define _Float64x flt64x_t
1618 #define _Float128x flt128x_t
```