

Matrix3000 LIBVM

数学库

接口手册

V1.2.3

(2022 年 2 月 18 日)

国防科技大学计算机学院 652 室

目录

1	Libvm 介绍	4
1.1	Libvm 支持的浮点标准	4
1.2	Libvm 使用的变量类型	4
1.3	Libvm 准确度	5
1.4	Libvm 实现的函数版本	5
1.5	Libvm 函数命名规范说明	7
1.6	Libvm 在 MT3000 上的非规格数支持情况	7
1.7	Libvm 在 MT3000 上的浮点异常处理	8
2	Libvm 向量数学函数	8
2.1	三角函数	8
2.1.1	sin 函数	8
2.1.2	cos 函数	10
2.1.3	tan 函数	11
2.1.4	sincos 函数	13
2.1.5	sinpi 函数	14
2.1.6	cospi 函数	15
2.1.7	sincospi 函数	15
2.2	反三角函数	16
2.2.1	asin 函数	16
2.2.2	acos 函数	18
2.2.3	atan 函数	20
2.2.4	atan2 函数	22
2.3	对数函数、指数函数、幂函数、开根号函数	23
2.3.1	log 函数	23
2.3.2	log10 函数	25
2.3.3	log2 函数	26
2.3.4	log1p 函数	26
2.3.5	exp 函数	27
2.3.6	exp2 函数	28
2.3.7	exp10 函数	29
2.3.8	expm1 函数	29
2.3.9	sqrt 函数	30
2.3.10	cbrt 函数	31
2.3.11	hypot 函数	32
2.3.12	pow 函数	33
2.4	双曲函数	34
2.4.1	sinh 函数	34
2.4.2	cosh 函数	35
2.4.3	tanh 函数	36
2.5	反双曲函数	37
2.5.1	asinh 函数	37
2.5.2	acosh 函数	38

2.5.3	atanh 函数	38
2.6	错误与 gamma 函数.....	39
2.6.1	lgamma 函数.....	39
2.6.2	gamma 函数.....	39
2.6.3	erf 函数.....	40
2.6.4	erfc 函数	40
2.7	浮点舍入（取整）函数.....	41
2.7.1	trunc 函数	41
2.7.2	floor 函数.....	42
2.7.3	ceil 函数	42
2.7.4	round 函数.....	43
2.7.5	rint 函数.....	43
2.8	基础函数.....	44
2.8.1	fma 函数	44
2.8.2	fmod 函数	45
2.8.3	ldexp 函数.....	45
2.8.4	frfrexp 函数.....	46
2.8.5	expfrexp 函数.....	46
2.8.6	ilogb 函数.....	47
2.8.7	modf 函数	47
2.8.8	abs 函数	47
2.8.9	fmax 函数.....	48
2.8.10	fmin 函数	49
2.8.11	fdim 函数	50
2.8.12	copysign 函数.....	51
2.8.13	nextafter 函数.....	52
2.9	除法与求倒数函数.....	52
2.9.1	fdiv 函数.....	52
2.9.2	frec 函数	53

1 Libvm 介绍

Libvm 数学库(A Library for Vector Math)是为 MT3000 开发的向量数学库，该数学库实现 C99 标准定义的实浮点数学函数的向量版本。此外，Libvm 还为 MT3000 缺少的向量浮点除法与求导指令提供对应的软件实现。

1.1 Libvm 支持的浮点标准

Libvm 函数库为 C99 标准定义的实浮点数学函数提供单精度和双精度浮点版本。

1.2 Libvm 使用的变量类型

Libvm 函数使用了 MT3000 向量 intrinsic 中定义 `lvector double`、`lvector float`、`lvector signed int` 来分别进行双精度、单精度浮点运算以及有符号整型运算，并将这些变量类型作为函数输入类型和返回值类型，其中：

1. 一个 `lvector double` 向量包含 16 个 `double` 类型 (64 bits) 双精度浮点数。
2. 一个 `lvector float` 向量包含 32 个 `float` 类型 (32 bits) 单精度浮点数。
3. 一个 `lvector signed int` 向量包含 32 个 `signed int` 类型 (32 bits) 的整型。

同时 Libvm 函数库内定义了 `lvector_double2`，`lvector_float2` 两种特殊向量类型。`lvector_double2` 类型存储两个 `lvector double` 变量，用于双精度情况下有两个返回值的数学函数计算中，以同时返回该函数的两个计算结果。其具体定义如下：

```
typedef struct {  
    lvector double x, y;  
} lvector_double2;
```

`lvector_float2` 类型存储两个 `lvector float` 变量，用于单精度情况下有两个返回值的数学函数计算中，以同时返回该函数的两个计算结果。其具体定义如

下:

```
typedef struct {
    lvector float x, y;
} lvector_float2;
```

1.3 Libvm 准确度

Libvm 函数库为所实现数学函数提供最大误差为 1.0ulp 或 0.5ulp 的版本（根据函数性质），并同时为大部分函数提供了最大误差为 3.5ulp 的版本。供程序员在使用函数时根据性能和准确度的需要进行选择，一般情况下 1.0ulp/0.5ulp 版本的准确度高性能低，3.5ulp 版本的性能高准确度低。

1.4 Libvm 实现的函数版本

Libvm 为各数学运算提供单、双精度版本向量函数实现（ldexp、expfrexp、ilogb 三个运算只提供双精度函数版本）。在单、双精度版本基础上，Libvm 为每个运算提供不同的准确度版本，包括 u10（函数的最大误差是 1.0ulp）、u05（函数的最大误差是 0.5ulp）以及 u35（函数的最大误差是 3.5ulp）版本。Libvm 实现的实浮点函数的向量版本如下表所示。

数学函数	双精度		单精度	
	u10 或 u05	u35	u10 或 u05	u35
三角函数				
sin	vm_sind16_u10	vm_sind16_u35	vm_sinf32_u10	vm_sinf32_u35
cos	vm_cosd16_u10	vm_cosd16_u35	vm_cosf32_u10	vm_cosf32_u35
sincos	vm_sincosd16_u10	vm_sincosd16_u35	vm_sincosf32_u10	vm_sincosf32_u35
sinpi	vm_sinpid16_u05	NA	vm_sinpif32_u05	NA
cospi	vm_cospid16_u05	NA	vm_cospif32_u05	NA
sincospi	vm_sincospid16_u05	NA	vm_sincospif32_u05	NA
tan	vm_tand16_u10	vm_tand16_u35	vm_tanf32_u10	vm_tanf32_u35
反三角函数				
asin	vm_asind16_u10	vm_asind16_u35	vm_asinf32_u10	vm_asinf32_u35
acos	vm_acosd16_u10	vm_acosd16_u35	vm_acosf32_u10	vm_acosf32_u35
atan	vm_atand16_u10	vm_atand16_u35	vm_atanf32_u10	vm_atanf32_u35
atan2	vm_atan2d16_u10	vm_atan2d16_u35	vm_atan2f32_u10	vm_atan2f32_u35
幂函数, 指数函数, 对数函数, 开方函数				
pow	vm_powd16_u10	NA	vm_powf32_u10	NA
log	vm_logd16_u10	vm_logd16_u35	vm_logf32_u10	vm_logf32_u35

log10	vm_log10d16_u10	NA	vm_log10f32_u10	NA
log2	vm_log2d16_u10	NA	vm_log2f32_u10	NA
log1p	vm_log1pd16_u10	NA	vm_log1pf32_u10	NA
exp	vm_expd16_u10	NA	vm_expf32_u10	NA
exp2	vm_exp2d16_u10	NA	vm_exp2f32_u10	NA
exp10	vm_exp10d16_u10	NA	vm_exp10f32_u10	NA
expm1	vm_expm1d16_u10	NA	vm_expm1f32_u10	NA
sqrt	vm_sqrt16_u05	vm_sqrt16_u35	vm_sqrtf32_u05	vm_sqrtf32_u35
cbrt	vm_cbrt16_u10	vm_cbrt16_u35	vm_cbrtf32_u10	vm_cbrtf32_u35
hypot	vm_hypot16_u05	vm_hypot16_u35	vm_hypotf32_u05	vm_hypotf32_u35
双曲函数				
sinh	vm_sinh16_u10	vm_sinh16_u35	vm_sinhf32_u10	vm_sinhf32_u35
cosh	vm_cosh16_u10	vm_cosh16_u35	vm_coshf32_u10	vm_coshf32_u35
tanh	vm_tanh16_u10	vm_tanh16_u35	vm_tanhf32_u10	vm_tanhf32_u35
反双曲函数				
asinh	vm_asinh16_u10	NA	vm_asinhf32_u10	NA
acosh	vm_acosh16_u10	NA	vm_acoshf32_u10	NA
atanh	vm_atanh16_u10	NA	vm_atanhf32_u10	NA
错误和 gamma 函数				
erf	vm_erf16_u10	NA	vm_erff32_u10	NA
erfc	vm_erfc16_u15	NA	vm_erfcf32_u15	NA
tgamma	vm_tgamma16_u10	NA	vm_tgammaf32_u10	NA
lgamma	vm_lgamma16_u10	NA	vm_lgammaf32_u10	NA
浮点舍入函数				
trunc	vm_trunc16	NA	vm_truncf32	NA
floor	vm_floor16	NA	vm_floorf32	NA
ceil	vm_ceil16	NA	vm_ceilf32	NA
round	vm_round16	NA	vm_roundf32	NA
rint	vm_rint16	NA	vm_rintf32	NA
基础函数				
fma	vm_fma16	NA	vm_fmaf32	NA
fmod	vm_fmod16	NA	vm_fmodf32	NA
ldexp	vm_ldexp16	NA	NA	NA
frfrexp	vm_frfrexp16	NA	vm_frfrexp32	NA
expfrexp	vm_expfrexp16	NA	NA	NA
ilogb	vm_ilogb16	NA	NA	NA
modf	vm_modf16	NA	vm_modff32	NA
fabs	vm_fabs16	NA	vm_fabsf32	NA
fmax	vm_fmax16	NA	vm_fmaxf32	NA
fmin	vm_fmin16	NA	vm_fminf32	NA
fdim	vm_fdim16	NA	vm_fdimf32	NA
copysign	vm_copysign16	NA	vm_copysignf32	NA
nextafter	vm_nextafter16	NA	vm_nextafterf32	NA

1.5 Libvm 函数命名规范说明

Libvm 的函数名形式如下：

`vm_xxx[d|f][16|32]_[u10|u05|u35]`

其中各个代表：

字符串“vm_”代表向量数学库 Libvm 的缩写。

字符串“xxx”代表该向量函数对应的数学运算，如 sin。

字符串“d”或“f”代表该向量函数所操作的浮点数的精度双精度为 d，单精度为 f。

数字 16 或 32 代表该向量函数的向量变量中浮点数的元素个数，双精度为 16 个，单精度为 32 个。

以_u 开头的字符串代表该向量函数的准确度(以 ulp 为单位)，包括 u10 (1.0ulp)，u05 (0.5ulp) 和 u35 (3.5ulp) 版本

1.6 Libvm 在 MT3000 上的非规格数支持情况

由于 MT3000 的浮点部件不支持非规格数，所以 Libvm 在 MT3000 上对非规格数的支持有限，大部分函数不支持非规格数。对于不支持非规格数的函数，Libvm 进行如下处理：

1. 如果 Libvm 函数输入为非规格数，则 Libvm 将该输入视作符号相同的浮点零(+0.0 或-0.0)，并参与后续的计算。
2. 如果 Libvm 函数计算结果真实值为非规格数，则 Libvm 将该结果舍入为符号相同的浮点零(+0.0 或-0.0)。

部分函数由于其运算的特殊性，Libvm 可以处理非规格数，使得函数计算结果符合相应的准确度标准。

libvm 中各函数对非规格数的支持情况，详见文档第 2 节各函数接口的说明

1.7 Libvm 在 MT3000 上的浮点异常处理

Libvm 函数在 MT3000 上不会对浮点异常进行处理, 需要使用者自行检查计算结果是否产生浮点异常。

2 Libvm 向量数学函数

2.1 三角函数

2.1.1 sin 函数

1. vm_sind16_u10

接口:

```
lvector double vm_sind16_u10(lvector double a);
```

说明:

双精度向量正弦函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 sin 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 +/-Inf: 返回 NaN

2. vm_sinf32_u10

接口:

```
lvector float vm_sinf32_u10(lvector float a);
```

说明:

单精度向量正弦函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 sin 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 $\pm\text{Inf}$: 返回 NaN

3. `vm_sind16_u35`

接口:

```
lvector double vm_sind16_u35(lvector double a);
```

说明:

双精度向量正弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 \sin 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 3.5ulp 。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 $\pm\text{Inf}$: 返回 NaN

4. `vm_sinf32_u35`

接口:

```
lvector float vm_sinf32_u35(lvector float a);
```

说明:

单精度向量正弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 \sin 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 3.5ulp 。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 $\pm\text{Inf}$: 返回 NaN

2.1.2 cos 函数

1. vm_cosd16_u10

接口:

```
lvector double vm_cosd16_u10(lvector double a);
```

说明:

双精度向量余弦函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 cos 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 +/-Inf: 返回 NaN

2. vm_cosf32_u10

接口:

```
lvector float vm_cosf32_u10(lvector float a);
```

说明:

单精度向量余弦函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 cos 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 +/-Inf: 返回 NaN

3. vm_cosd16_u35

接口:

```
lvector double vm_cosd16_u35(lvector double a);
```

说明:

双精度向量余弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `cos` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持：

该函数可以正确处理输入为非规格数的情况

特殊值情况：

输入为 `NaN`：返回 `NaN`

输入为 `+/-Inf`：返回 `NaN`

4. `vm_cosf32_u35`

接口：

```
lvector float vm_cosf32_u35(lvector float a);
```

说明：

单精度向量余弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `cos` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持：

该函数可以正确处理输入为非规格数的情况

特殊值情况：

输入为 `NaN`：返回 `NaN`

输入为 `+/-Inf`：返回 `NaN`

2.1.3 `tan` 函数

1. `vm_tand16_u10`

接口：

```
lvector double vm_tand16_u10(lvector double a);
```

说明：

双精度向量正切函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `tan` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 +/-Inf: 返回 NaN

2. **vm_tanf32_u10**

接口:

```
lvector float vm_tanf32_u10(lvector float a);
```

说明:

单精度向量正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `tan` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 +/-Inf: 返回 NaN

3. **vm_tand16_u35**

接口:

```
lvector double vm_tand16_u35(lvector double a);
```

说明:

双精度向量正切函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `tan` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN: 返回 NaN

输入为 +/-Inf: 返回 NaN

4. `vm_tanf32_u35`

接口:

```
lvector float vm_tanf32_u35(lvector float a);
```

说明:

单精度向量正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `tan` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 `NaN`: 返回 `NaN`

输入为 `+/-Inf`: 返回 `NaN`

2.1.4 `sincos` 函数

1. `vm_sincosd16_u10`

接口:

```
lvector_double2 vm_sincosd16_u10(lvector double a);
```

说明:

该函数同时计算向量变量 `a` 每个元素的 `sin` 操作与 `cos` 操作的结果, 并把结果存储在 `lvector_double2` 类型变量的 `x` 与 `y` 元素中并返回结果。该函数的最大误差是 `1.0ulp`。

2. `vm_sincosf32_u10`

接口:

```
lvector_float2 vm_sincosf32_u10(lvector float a);
```

说明:

单精度向量 `sincos` 函数。该函数同时计算向量变量 `a` 每个元素的 `sin` 运算与 `cos` 运算的结果, 并把结果存储在 `lvector_float2` 类型变量的 `x` 与 `y` 元素中并返回结果。该函数的最大误差是 `1.0ulp`。

3. `vm_sincosd16_u35`

接口:

```
lvector_double2 vm_sincosd16_u35(lvector_double a);
```

说明:

双精度向量 sincos 函数。该函数同时计算向量变量 a 每个元素的 sin 运算与 cos 运算的结果, 并把结果存储在 lvector_double2 类型变量的 x 与 y 元素中并返回结果。该函数的最大误差是 3.5ulp。

4. vm_sincosf32_u35

接口:

```
lvector_float2 vm_sincosf32_u35(lvector_float a);
```

说明:

单精度向量 sincos 函数。该函数同时计算向量变量 a 每个元素的 sin 运算与 cos 运算的结果, 并把结果存储在 lvector_float2 类型变量的 x 与 y 元素中并返回结果。该函数的最大误差是 3.5ulp。

2.1.5 sinpi 函数

1. vm_sinpid16_u05

接口:

```
lvector_double vm_sinpid16_u05(lvector_double a);
```

说明:

双精度向量 sinpi 函数。该函数计算 lvector_double 向量类型变量 a 的每一个元素的 $\sin(\pi a)$ 值, 并将结果存储于 lvector_double 变量中返回。该函数最大误差为 0.506ulp。

2. vm_sinpif32_u05

接口:

```
lvector_float vm_sinpif32_u05(lvector_float a);
```

说明:

单精度向量 sinpi 函数。该函数计算 lvector_float 向量类型变量 a 的每一个元素的 $\sin(\pi a)$ 值, 并将结果存储于 lvector_float 变量中返回。该函数最大误差为 0.506ulp。

2.1.6 cospi 函数

1. vm_cospid16_u05

接口:

```
lvector double vm_cospid16_u05(lvector double a);
```

说明:

双精度向量 cospi 函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 $\cos(\pi a)$ 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 0.506ulp。

2. vm_cospif32_u05

接口:

```
lvector float vm_cospif32_u05(lvector float a);
```

说明:

单精度向量 cospi 函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 $\cos(\pi a)$ 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 0.506ulp。

2.1.7 sincospi 函数

1. vm_sincospid16_u05

接口:

```
lvector_double2 vm_sincospid16_u05(lvector double a);
```

说明:

双精度向量 sincospi 函数。该函数同时计算向量变量 a 每个元素的 $\sin(\pi a)$ 与 $\cos(\pi a)$ 的结果, 并把结果存储在 lvector_double2 类型变量的 x 与 y 元素中并返回结果。该函数的最大误差是 1.0ulp。

2. vm_sincospif32_u05

接口:

```
lvector_float2 vm_sincospif32_u05(lvector float a);
```

说明:

单精度向量 sincospi 函数。该函数同时计算向量变量 a 每个元素的 $\sin(\pi a)$ 与 $\cos(\pi a)$ 的结果，并把结果存储在 lvector_float2 类型变量的 x 与 y 元素中并返回结果。该函数的最大误差是 1.0ulp。

3. vm_sincospid16_u35

接口：

```
lvector_double2 vm_sincospid16_u35(lvector_double a);
```

说明：

双精度向量 sincospi 函数。该函数同时计算向量变量 a 每个元素的 $\sin(\pi a)$ 与 $\cos(\pi a)$ 的结果，并把结果存储在 lvector_double2 类型变量的 x 与 y 元素中并返回结果。该函数的最大误差是 3.5ulp。

4. vm_sincospif32_u35

接口：

```
lvector_float2 vm_sincospif32_u35(lvector_float a);
```

说明：

单精度向量 sincospi 函数。该函数同时计算向量变量 a 每个元素的 $\sin(\pi a)$ 与 $\cos(\pi a)$ 的结果，并把结果存储在 lvector_float2 类型变量的 x 与 y 元素中并返回结果。该函数的最大误差是 3.5ulp。

2.2 反三角函数

2.2.1 asin 函数

1. vm_asind16_u10

接口：

```
lvector_double vm_asind16_u10(lvector_double a);
```

说明：

双精度向量反正弦函数。该函数计算 lvector_double 向量类型变量 a 的每一个元素的 asin 值，并将结果存储于 lvector_double 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持：

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外: 返回 NaN

2. `vm_asinf32_u10`

接口:

```
lvector float vm_asinf32_u10(lvector float a);
```

说明:

单精度向量反正弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `asin` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外: 返回 NaN

3. `vm_asind16_u35`

接口:

```
lvector double vm_asind16_u35(lvector double a);
```

说明:

双精度向量反正弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `asin` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外: 返回 NaN

4. `vm_asinf32_u35`

接口:

```
lvector float vm_asinf32_u35(lvector float a);
```

说明:

单精度向量反正弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `asin` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 3.5ulp 。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外: 返回 NaN

2.2.2 `acos` 函数

1. `vm_acosd16_u10`

接口:

```
lvector double vm_acosd16_u10(lvector double a);
```

说明:

双精度向量反余弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `acos` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 1.0ulp 。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外:返回 NaN

2. `vm_acosf32_u10`

接口:

```
lvector float vm_acosf32_u10(lvector float a);
```

说明:

单精度向量反余弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `acos` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外:返回 NaN

3. `vm_acosd16_u35`

接口:

```
lvector double vm_acosd16_u35(lvector double a);
```

说明:

双精度向量反余弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `acos` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外:返回 NaN

4. `vm_acosf32_u35`

接口:

```
lvector float vm_acosf32_u35(lvector float a);
```

说明:

单精度向量反余弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `acos` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入在 $[-1.0, +1.0]$ 之外: 返回 NaN

2.2.3 atan 函数

1. vm_atand16_u10

接口:

```
lvector double vm_atand16_u10(lvector double a);
```

说明:

双精度向量反正切函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `atan` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入为 $\pm \text{Inf}$: 返回 $\pm \pi/2$ 或者 $-\pi/2$

2. vm_atanf32_u10

接口:

```
lvector float vm_atanf32_u10(lvector float a);
```

说明:

单精度向量反正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `atan` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入为 $\pm \text{Inf}$: 返回 $\pm \pi/2$ 或者 $-\pi/2$

3. `vm_atand16_u35`

接口:

```
lvector double vm_atand16_u35(lvector double a);
```

说明:

双精度向量反正切函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `atan` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

非规格数支持:

该函数可以正确处理输入为非规格数的情况

特殊值情况:

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入为 $\pm \text{Inf}$: 返回 $\pm \pi/2$ 或者 $-\pi/2$

4. `vm_atanf32_u35`

接口:

```
lvector float vm_atanf32_u35(lvector float a);
```

说明:

单精度向量反正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `atan` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误

差为 3.5ulp。

非规格数支持：

该函数可以正确处理输入为非规格数的情况

特殊值情况：

输入为 NaN : 返回 NaN

输入为 ± 0.0 : 返回 ± 0.0

输入为 $\pm \text{Inf}$: 返回 $\pm \pi/2$ 或者 $-\pi/2$

2.2.4 atan2 函数

1. vm_atan2d16_u10

接口：

```
lvector double vm_atan2d16_u10(lvector double y, lvector double x);
```

说明：

双精度向量双操作数反正切函数，该函数针对 lvector double 类型输入 x 和 y，计算 x 和 y 每个元素 (y_i/x_i) 的 atan 值。并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_atan2f32_u10

接口：

```
lvector float vm_atan2f32_u10(lvector float y, lvector float x);
```

说明：

单精度向量双操作数反正切函数该，该函数针对 lvector float 类型输入 x 和 y，计算 x 和 y 每个元素 (y_i/x_i) 的 atan 值。并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

3. vm_atan2d16_u35

接口：

```
lvector double vm_atan2d16_u35(lvector double y, lvector double x);
```

说明：

双精度向量双操作数反正切函数该，该函数针对 lvector double 类型输入 x 和 y，计算 x 和 y 每个元素 (y_i/x_i) 的 atan 值。并将结果存储于 lvector

double 变量中返回。该函数最大误差为 3.5ulp。

4. `vm_atan2f32_u35`

接口:

```
lvector float vm_atan2f32_u35(lvector float y, lvector float x);
```

说明:

单精度向量双操作数反正切函数该, 该函数针对 `lvector float` 类型输入 `x` 和 `y`, 计算 `x` 和 `y` 每个元素 (y_i/x_i) 的 `atan` 值。并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 3.5ulp。

2.3 对数函数、指数函数、幂函数、开根号函数

2.3.1 `log` 函数

1. `vm_logd16_u10`

接口:

```
lvector double vm_logd16_u10(lvector double a);
```

说明:

双精度向量对数函数, 该对数函数以自然对数 e 为底数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `log` 值 (以 e 为底数), 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持:

有限的非规格数支持, 输入为负非规格数可返回正确的 NaN 值, 输入为正非规格数会因硬件不支持非规格数而返回 `-Inf`。

特殊值情况:

输入为 NaN	: 返回 NaN
输入为 1.0	: 返回+0.0
输入为+Inf	: 返回+Inf
输入为+/-0	: -Inf
输入为负数	: 返回 NaN

2. `vm_logf32_u10`

接口:

```
lvector float vm_logf32_u10(lvector float a);
```

说明:

单精度向量对数函数, 该对数函数以自然对数 e 为底数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 \log 值 (以 e 为底数), 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 1.0ulp 。

非规格数支持:

有限的非规格数支持, 输入为负非规格数可返回正确的 NaN 值, 输入为正非规格数会因硬件不支持非规格数而返回 $-\text{Inf}$ 。

特殊值情况:

输入为 NaN	: 返回 NaN
输入为 1.0	: 返回+0.0
输入为+Inf	: 返回+Inf
输入为+/-0	: $-\text{Inf}$
输入为负数	: 返回 NaN

3. `vm_logd16_u35`

接口:

```
lvector double vm_logd16_u35(lvector double a);
```

说明:

双精度向量对数函数, 该对数函数以自然对数 e 为底数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 \log 值 (以 e 为底数), 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 3.5ulp 。

非规格数支持:

有限的非规格数支持, 输入为负非规格数可返回正确的 NaN 值, 输入为正非规格数会因硬件不支持非规格数而返回 $-\text{Inf}$ 。

特殊值情况:

输入为 NaN	: 返回 NaN
输入为 1.0	: 返回+0.0
输入为+Inf	: 返回+Inf

输入为+/-0 : -Inf

输入为负数 : 返回 NaN

4. `vm_logf32_u35`

接口:

```
lvector float vm_logf32_u35(lvector float a);
```

说明:

单精度向量对数函数, 该对数函数以自然对数 e 为底数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 \log 值 (以 e 为底数), 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 3.5ulp 。

非规格数支持:

有限的非规格数支持, 输入为负非规格数可返回正确的 NaN 值, 输入为正非规格数会因硬件不支持非规格数而返回 -Inf。

特殊值情况:

输入为 NaN : 返回 NaN

输入为 1.0 : 返回+0.0

输入为+Inf : 返回+Inf

输入为+/-0 : -Inf

输入为负数 : 返回 NaN

2.3.2 `log10` 函数

1. `vm_log10d16_u10`

接口:

```
lvector double vm_log10d16_u10(lvector double a);
```

说明:

双精度向量对数函数, 该对数函数以 10 为底数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 \log 值 (以 10 为底数), 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 1.0ulp 。

2. `vm_log10f32_u10`

接口:

```
lvector float vm_log10f32_u10(lvector float a);
```

说明:

单精度向量对数函数。该对数函数以 10 为底数，并且该函数计算 lvector float 向量类型变量 a 的每一个元素的 log 值（以 10 为底数），并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.3.3 log2 函数

1. vm_log2d16_u10

接口:

```
lvector double vm_log2d16_u10(lvector double a);
```

说明:

双精度向量对数函数，该对数函数以 2 为底数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 log 值（以 2 为底数），并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_log2f32_u10

接口:

```
lvector float vm_log2f32_u10(lvector float a);
```

说明:

单精度向量对数函数，该对数函数以 2 为底数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 log 值（以 2 为底数），并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.3.4 log1p 函数

1. vm_log1pd16_u10

接口:

```
lvector double vm_log1pd16_u10(lvector double a);
```

说明:

双精度向量对数函数，该对数函数以自然对数 e 为底数。该函数计算 lvector double 向量类型变量 a 的每一个元素 a_i 加 1.0 的 log 值 ($\log(a_i+1.0)$)，以 e 为

底数), 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_loglpf32_u10

接口:

```
lvector float vm_loglpf32_u10(lvector float a);
```

说明:

单精度向量对数函数, 该对数函数以自然对数 e 为底数。该函数计算 lvector float 向量类型变量 a 的每一个元素 ai 加 1.0 的 log 值($\log(ai+1.0)$), 以 e 为底数), 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.3.5 exp 函数

1. vm_expdl6_u10

接口:

```
lvector double vm_expdl6_u10(lvector double a);
```

说明:

双精度向量指数函数, 该指数函数以自然对数 e 为底数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 exp 值 (以 e 为底数), 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

非规格数支持:

不支持非规格数, 当输入为 $[-7.444401e+02, -7.083964e+02]$ 之间的值时, 由于计算结果真实值为非规格数, Libvm 返回+0.0

特殊值情况:

输入为 NaN	: 返回 NaN
输入为+Inf	: 返回+Inf
输入为-Inf	: 返回+0.0
结果下溢	: 返回+0.0
结果上溢	: 返回+Inf

2. vm_exp32_u10

接口:

```
lvector float vm_exp32_u10(lvector float a);
```

说明:

单精度向量指数函数, 该指数函数以自然对数 e 为底数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `exp` 值 (以 e 为底数), 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 1.0ulp 。

非规格数支持 :

不支持非规格数数: 当输入为 $[-103.278929903, -87.3365448704]$ 之间的值时, 由于计算结果真实值为非规格数, Libvm 返回 $+0.0$

特殊值情况:

输入为 NaN	: 返回 NaN
输入为 +Inf	: 返回 +Inf
输入为 -Inf	: 返回 $+0.0$
结果下溢	: 返回 $+0.0$
结果上溢	: 返回 +Inf

2.3.6 exp2 函数

1. vm_exp2d16_u10

接口:

```
lvector double vm_exp2d16_u10(lvector double a);
```

说明:

双精度向量指数函数, 该指数函数以 2 为底数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `exp` 值 (以 2 为底数), 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 1.0ulp 。

2. vm_exp2f32_u10

接口:

```
lvector float vm_exp2f32_u10(lvector float a);
```

说明:

单精度向量指数函数, 该指数函数以 2 为底数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `exp` 值 (以 2 为底数), 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 1.0ulp 。

2.3.7 exp10 函数

1. vm_exp10d16_u10

接口:

```
lvector double vm_exp10d16_u10(lvector double a);
```

说明:

双精度向量指数函数, 该指数函数以 10 为底数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 exp 值(以 10 为底数), 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_exp10f32_u10

接口:

```
lvector float vm_exp10f32_u10(lvector float a);
```

说明:

单精度向量指数函数, 该指数函数以 10 为底数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 exp 值(以 e 为底数), 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.3.8 expm1 函数

1. vm_expm1d16_u10

接口:

```
lvector double vm_expm1d16_u10(lvector double a);
```

说明:

双精度向量指数函数, 该指数函数以自然对数 e 为底数。该函数计算 lvector double 向量类型变量 a 的每一个元素 $(\exp(a_i) - 1.0)$ 的值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_expm1f32_u10

接口:

```
lvector float vm_expm1f32_u10(lvector float a);
```

说明:

单精度向量指数函数, 该指数函数以自然对数 e 为底数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素 ($\exp(a_i) - 1.0$) 的值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 1.0ulp 。

2.3.9 sqrt 函数

1. `vm_sqrtd16`

接口:

```
lvector double vm_sqrtd16(lvector double a);
```

说明:

双精度向量平方根函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `sqrt` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 0.5001ulp 。

2. `vm_sqrtf32`

接口:

```
lvector float vm_sqrtf32(lvector float a);
```

说明:

单精度向量平方根函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `sqrt` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 0.5001ulp 。

3. `vm_sqrtd16_u05`

接口:

```
lvector double vm_sqrtd16_u05(lvector double a);
```

说明:

双精度向量平方根函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `sqrt` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 0.5001ulp 。

4. `vm_sqrtf32_u05`

接口:

```
lvector float vm_sqrtf32_u05(lvector float a);
```

说明:

单精度向量平方根函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `sqrt` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `0.5001ulp`。

5. `vm_sqrt16_u35`

接口:

```
lvector double vm_sqrt16_u35(lvector double a);
```

说明:

双精度向量平方根函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `sqrt` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

6. `vm_sqrtf32_u35`

接口:

```
lvector float vm_sqrtf32_u35(lvector float a);
```

说明:

单精度向量平方根函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `sqrt` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `3.5ulp`。

2.3.10 `cbirt` 函数

1. `vm_cbirt16_u10`

接口:

```
lvector double vm_cbirt16_u10(lvector double a);
```

说明:

双精度向量立方根函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `cbirt` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

2. `vm_cbirtf32_u10`

接口:

```
lvector float vm_cbrtf32_u10(lvector float a);
```

说明:

单精度向量立方根函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `cbrt` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 1.0ulp。

3. `vm_cbrtd16_u35`

接口:

```
lvector double vm_cbrtd16_u35(lvector double a);
```

说明:

双精度向量立方根函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `cbrt` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 3.5ulp

4. `vm_cbrtf32_u35`

接口:

```
lvector float vm_cbrtf32_u35(lvector float a);
```

说明:

单精度向量立方根函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `cbrt` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 3.5ulp。

2.3.11 `hypot` 函数

1. `vm_hypotd16_u05`

接口:

```
lvector double vm_hypotd16_u05(lvector double x, lvector double y);
```

说明:

双精度向量 2D Euclidian distance 函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `hypot` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 0.5ulp。

2. `vm_hypotf32_u05`

接口:

```
lvector float vm_hypotf32_u05(lvector float x, lvector float y);
```

说明:

单精度向量 2D Euclidian distance 函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 hypot 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 0.5ulp。

3. vm_hypotd16_u35

接口:

```
lvector double vm_hypotd16_u35(lvector double x, lvector double y);
```

说明:

双精度向量 2D Euclidian distance 函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 hypot 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 3.5ulp。

4. vm_hypotf32_u35

接口:

```
lvector float vm_hypotf32_u35(lvector float x, lvector float y);
```

说明:

单精度向量 2D Euclidian distance 函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 hypot 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 3.5ulp。

2.3.12 pow 函数

1. vm_powd16_u10

接口:

```
lvector double vm_powd16_u10(lvector double x, lvector double y);
```

说明:

双精度向量 power 函数。该函数计算 lvector double 向量类型变量 x 和 y 对应的每一个元素 x_i 和 y_i 的 $\text{pow}(x_i, y_i)$ 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_powf32_u10

接口:

```
lvector float vm_powf32_u10(lvector float x, lvector float y);
```

说明:

单精度向量 power 函数。该函数计算 lvector float 向量类型变量 x 和 y 对应的每一个元素 x_i 和 y_i 的 $\text{pow}(x_i, y_i)$ 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.4 双曲函数

2.4.1 sinh 函数

1. vm_sinhd16_u10

接口:

```
lvector double vm_sinhd16_u10(lvector double a);
```

说明:

双精度向量双曲正弦函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 sinh 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_sinhf32_u10

接口:

```
lvector float vm_sinhf32_u10(lvector float a);
```

说明:

单精度向量双曲正弦函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 sinh 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

3. vm_sinhd16_u35

接口:

```
lvector double vm_sinhd16_u35(lvector double a);
```

说明:

双精度向量双曲正弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `sinh` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

4. `vm_sinhf32_u35`

接口：

```
lvector float vm_sinhf32_u35(lvector float a);
```

说明：

单精度向量双曲正弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `sinh` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `3.5ulp`。

2.4.2 `cosh` 函数

1. `vm_coshd16_u10`

接口：

```
lvector double vm_coshd16_u10(lvector double a);
```

说明：

双精度向量双曲余弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `cosh` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

2. `vm_coshf32_u10`

接口：

```
lvector float vm_coshf32_u10(lvector float a);
```

说明：

单精度向量双曲余弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `cosh` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

3. `vm_coshd16_u35`

接口：

```
lvector double vm_coshd16_u35(lvector double a);
```

说明:

双精度向量双曲余弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `cosh` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `3.5ulp`。

4. `vm_coshf32_u35`

接口:

```
lvector float vm_coshf32_u35(lvector float a);
```

说明:

单精度向量双曲余弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `cosh` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `3.5ulp`。

2.4.3 `tanh` 函数

1. `vm_tanhd16_u10`

接口:

```
lvector double vm_tanhd16_u10(lvector double a);
```

说明:

双精度向量双曲正切函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `tanh` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

2. `vm_tanhf32_u10`

接口:

```
lvector float vm_tanhf32_u10(lvector float a);
```

说明:

单精度向量双曲正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `tanh` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

3. `vm_tanhd16_u35`

接口:

```
lvector double vm_tanhd16_u35(lvector double a);
```

说明:

双精度向量双曲正切函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `tanh` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 3.5ulp。

4. `vm_tanhf32_u35`

接口:

```
lvector float vm_tanhf32_u35(lvector float a);
```

说明:

单精度向量双曲正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `tanh` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 3.5ulp。

2.5 反双曲函数

2.5.1 `asinh` 函数

1. `vm_asinhd16_u10`

接口:

```
lvector double vm_asinhd16_u10(lvector double a);
```

说明:

双精度向量反双曲正弦函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `asinh` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 1.0ulp。

2. `vm_asinhf32_u10`

接口:

```
lvector float vm_asinhf32_u10(lvector float a);
```

说明:

单精度向量反双曲正弦函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `asinh` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最

大误差为 1.0ulp。

2.5.2 **acosh 函数**

1. **vm_acoshd16_u10**

接口:

```
lvector double vm_acoshd16_u10(lvector double a);
```

说明:

双精度向量反双曲余弦函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 acosh 值，并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. **vm_acoshf32_u10**

接口:

```
lvector float vm_acoshf32_u10(lvector float a);
```

说明:

单精度向量反双曲余弦函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 acosh 值，并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.5.3 **atanh 函数**

1. **vm_atanhd16_u10**

接口:

```
lvector double vm_atanhd16_u10(lvector double a);
```

说明:

双精度向量反双曲正切函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 atanh 值，并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. **vm_atanhf32_u10**

接口:

```
lvector float vm_atanhf32_u10(lvector float a);
```

说明:

单精度向量反双曲正切函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `atanh` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

2.6 错误与 gamma 函数

2.6.1 lgamma 函数

1. `vm_lgammad16_u10`

接口:

```
lvector double vm_lgammad16_u10(lvector double a);
```

说明:

双精度向量 `log gamma` 函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `lgamma` 值, 并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

2. `vm_lgammaf32_u10`

接口:

```
lvector float vm_lgammaf32_u10(lvector float a);
```

说明:

单精度向量 `log gamma` 函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `lgamma` 值, 并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

2.6.2 gamma 函数

1. `vm_tgammad16_u10`

接口:

```
lvector double vm_tgammad16_u10(lvector double a);
```

说明:

双精度向量 `gamma` 函数。该函数计算 `lvector double` 向量类型变量 `a` 的每

一个元素的 `tgamma` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

2. `vm_tgammaf32_u10`

接口：

```
lvector float vm_tgammaf32_u10(lvector float a);
```

说明：

单精度向量 `gamma` 函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `tgamma` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

2.6.3 `erf` 函数

1. `vm_erfd16_u10`

接口：

```
lvector double vm_erfd16_u10(lvector double a);
```

说明：

双精度向量 `error` 函数。该函数计算 `lvector double` 向量类型变量 `a` 的每一个元素的 `erf` 值，并将结果存储于 `lvector double` 变量中返回。该函数最大误差为 `1.0ulp`。

2. `vm_erff32_u10`

接口

```
lvector float vm_erff32_u10(lvector float a);
```

说明：

单精度向量 `error` 函数。该函数计算 `lvector float` 向量类型变量 `a` 的每一个元素的 `erf` 值，并将结果存储于 `lvector float` 变量中返回。该函数最大误差为 `1.0ulp`。

2.6.4 `erfc` 函数

1. `vm_erfcd16_u15`

接口：


```
lvector double vm_erfcd16_u15(lvector double a);
```

说明:

双精度向量 complementary error 函数。该函数计算 lvector double 向量类型变量 a 的每一个元素的 erfc 值, 并将结果存储于 lvector double 变量中返回。该函数最大误差为 1.0ulp。

2. vm_erfcf32_u15

接口:

```
lvector float vm_erfcf32_u15(lvector float a);
```

说明:

单精度向量 complementary error 函数。该函数计算 lvector float 向量类型变量 a 的每一个元素的 erfc 值, 并将结果存储于 lvector float 变量中返回。该函数最大误差为 1.0ulp。

2.7 浮点舍入（取整）函数

2.7.1 trunc 函数

1. vm_truncd16

接口:

```
lvector double vm_truncd16(lvector double a);
```

说明:

双精度向量浮点舍入（取整）函数, 该舍入函数使用“向零舍入”模式将 lvector double 向量类型变量 a 的每一个浮点数舍入为浮点整数, 并将结果存储于 lvector double 变量中返回。该函数是 math.h 中 trunc 函数的向量版本。

2. vm_truncf32

接口:

```
lvector float vm_truncf32(lvector float a);
```

说明:

单精度向量浮点舍入（取整）函数, 该舍入函数使用“向零舍入”模式将 lvector float 向量类型变量 a 的每一个元素舍入为浮点整数, 并将结果存储于

lvector float 变量中返回。该函数是 math.h 中 truncf 函数的向量版本。

2.7.2 floor 函数

1. vm_floord16

接口:

```
lvector double vm_floord16(lvector double a);
```

说明:

双精度向量浮点舍入（取整）函数，该舍入函数使用“向负无穷大舍入”模式将 lvector double 向量类型变量 a 的每一个浮点数舍入为浮点整数，并将结果存储于 lvector double 变量中返回。该函数是 math.h 中 floor 函数的向量版本。

2. vm_floorf32

接口:

```
lvector float vm_floorf32(lvector float a);
```

说明:

单精度向量浮点舍入（取整）函数，该舍入函数使用“向负无穷大舍入”模式将 lvector float 向量类型变量 a 的每一个元素舍入为浮点整数，并将结果存储于 lvector float 变量中返回。该函数是 math.h 中 floorf 函数的向量版本。

2.7.3 ceil 函数

1. vm_ceild16

接口:

```
lvector double vm_ceild16(lvector double a);
```

说明:

双精度向量浮点舍入（取整）函数，该舍入函数使用“向正无穷大舍入”模式将 lvector double 向量类型变量 a 的每一个浮点数舍入为浮点整数，并将结果存储于 lvector double 变量中返回。该函数是 math.h 中 ceil 函数的向量版本。

2. vm_ceilf32

接口:

```
lvector float vm_ceilf32(lvector float a);
```

说明:

单精度向量浮点舍入（取整）函数，该舍入函数使用“向正无穷大舍入”模式将 `lvector float` 向量类型变量 `a` 的每一个元素舍入为浮点整数，并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `ceilf` 函数的向量版本。

2.7.4 round 函数

1. vm_roundd16

接口:

```
lvector double vm_roundd16(lvector double a);
```

说明:

双精度向量浮点舍入（取整）函数，该舍入函数使用就近舍入模式（偏向无穷大）将 `lvector double` 向量类型变量 `a` 的每一个浮点数舍入为浮点整数，并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `round` 函数的向量版本。

2. vm_roundf32

接口:

```
lvector float vm_roundf32(lvector float a);
```

说明:

单精度向量浮点舍入（取整）函数，该舍入函数使用就近舍入模式（偏向无穷大）将 `lvector float` 向量类型变量 `a` 的每一个元素舍入为浮点整数，并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `roundf` 函数的向量版本。

2.7.5 rint 函数

1. vm_rintd16

接口:

```
lvector double vm_rintd16(lvector double a);
```

说明:

双精度向量浮点舍入（取整）函数，该舍入函数使用就近舍入模式（偏向偶数）将 `lvector double` 向量类型变量 `a` 的每一个浮点数舍入为浮点整数，并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `rint` 函数的向量版本。

2. `vm_rintf32`

接口:

```
lvector float vm_rintf32(lvector float a);
```

说明:

单精度向量浮点舍入（取整）函数，该舍入函数使用就近舍入模式（偏向偶数）将 `lvector float` 向量类型变量 `a` 的每一个元素舍入为浮点整数，并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `rintf` 函数的向量版本。

2.8 基础函数

2.8.1 `fma` 函数

1. `vm_fmadd16`

接口:

```
lvector double vm_fmadd16(lvector double a, lvector double b,  
lvector double c);
```

说明:

双精度向量乘加函数，该函数针对输入的 `lvector double` 向量变量 `a`、`b`、`c`，将其对应的各元素 `ai`、`bi`、`ci` 进行乘加运算 ($a_i * b_i + c_i$)，并将结果存储于 `lvector double` 变量中返回，该函数内部实行使用 MT3000 中的乘加指令。该函数是 `math.h` 中 `fma` 函数的向量版本。

2. `vm_fmaf32`

接口:

```
lvector float vm_fmaf32(lvector float a, lvector float b, lvector  
float c);
```

说明:

单精度向量乘加函数, 该函数针对输入的 `lvector float` 向量变量 `a`、`b`、`c`, 将其对应的各元素 `ai`、`bi`、`ci` 进行乘加运算($ai*bi+ci$), 并将结果存储于 `lvector float` 变量中返回, 该函数内部实行使用 MT3000 中的乘加指令。该函数是 `math.h` 中 `fmaf` 函数的向量版本。

2.8.2 fmod 函数

1. vm_fmodd16

接口:

```
lvector double vm_fmodd16(lvector double a, lvector double b);
```

说明:

双精度向量求余函数, 该函数求取 `lvector double` 向量类型变量 `a` 的每个浮点数元素的余数, 并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `fmod` 函数的向量版本

2. vm_fmodf32

接口:

```
lvector float vm_fmodf32(lvector float a, lvector float b);
```

说明:

单精度向量求余函数, 该函数求取 `lvector float` 向量类型变量 `a` 的每个浮点数元素的余数, 并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `fmodf` 函数的向量版本。

2.8.3 ldexp 函数

1. vm_ldexpd16

接口:

```
lvector double vm_ldexpd16(lvector double a, lvector signed int b);
```

说明:

双精度向量 `ldexp` 函数, 该函数针对输入的 `lvector double` 类型向量 `a` 和向量 `b`, 计算两者对应的各个元素 `ai` 和 `bi` 的 $(ai * 2^{bi})$ 值, 并将结果存储

于 `lvector double` 变量中返回。该函数是 `math.h` 中 `ldexp` 函数的向量版本(该函数只有双精度版本)。

2.8.4 frfrexp 函数

1. `vm_frfrexpdl6`

接口:

```
lvector double vm_frfrexpdl6(lvector double a);
```

说明:

双精度向量 `frfrexp` 函数(求浮点数的尾数), 该函数求取输入的 `lvector double a` 各个浮点元素的尾数, 并将结果存储于 `lvector double` 变量中返回。该函数(返回尾数)与 2.8.5 中的 `expfrexp` 函数(返回指数)一起是 `math.h` 中 `frexp` 函数的向量版本。

2. `vm_frfrexpfloat32`

接口:

```
lvector float vm_frfrexpfloat32(lvector float a);
```

说明:

单精度向量 `frfrexp` 函数(求浮点数的尾数), 该函数求取输入的 `lvector float a` 各个浮点元素的尾数, 并将结果存储于 `lvector float` 变量中返回。该函数(返回尾数)是 `math.h` 中的 `frexpf` 函数部分功能的向量版本。

2.8.5 expfrexp 函数

1. `vm_expfrexpdl6`

接口:

```
lvector signed int vm_expfrexpdl6(lvector double a);
```

说明:

双精度向量 `expfrexp` 函数(求浮点数的指数), 该函数求取输入的 `lvector double a` 各个浮点元素的指数, 并将整型结果存储于 `lvector signed int` 变量中返回(该函数只有双精度版本)。该函数(返回指数)与 2.8.4 中的 `frfrexp` 函数(返回尾数)一起是 `math.h` 中 `frexp` 函数的向量版本。

2.8.6 ilogb 函数

1. vm_ilogbd16

接口:

```
lvector signed int vm_ilogbd16(lvector double a);
```

说明:

双精度向量 `ilogb` 函数(求浮点数的指数), 该函数求取输入的 `lvector double a` 各个浮点元素的指数, 并将整型结果存储于 `lvector signed int` 变量中返回 (该函数只有双精度版本)。该函数是 `math.h` 中 `ilogb` 函数的向量版本。

2.8.7 modf 函数

1. vm_modfd16

接口:

```
lvector_double2 vm_modfd16(lvector double a);
```

说明:

双精度向量 `modf` 函数, 该函数将输入的 `lvector double` 向量类型变量 `a` 的每一个浮点元素拆分成整数和小数部分, 并将结果分别存储于 `lvector_double2` 变量的 `x` 和 `y` 元素中返回。该函数是 `math.h` 中 `modf` 函数的向量版本。

2. vm_modff32

接口:

```
lvector_float2 vm_modff32(lvector float a);
```

说明:

单精度向量 `modf` 函数, 该函数将输入的 `lvector float` 向量类型变量 `a` 的每一个浮点元素拆分成整数和小数部分, 并将结果分别存储于 `lvector_float2` 变量的 `x` 和 `y` 元素中返回。该函数是 `math.h` 中 `modff` 函数的向量版本。

2.8.8 abs 函数

1. vm_fabsd16

接口:

```
lvector double  vm_fabsd16(lvector double a);
```

说明:

双精度向量求绝对值函数, 该函数求取输入的 `lvector double` 向量类型变量 `a` 的每一个浮点数的绝对值, 并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `fabs` 函数的向量版本。

2. `vm_fabsf32`

接口:

```
lvector float  vm_fabsf32(lvector float a);
```

说明:

单精度向量求绝对值函数, 该函数求取输入的 `lvector float` 向量类型变量 `a` 的每一个浮点数的绝对值, 并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `fabsf` 函数的向量版本。

举例:

假设 `lvector double` 元素个数为 2

输入:

```
lvector double a = {a1, a2 };
```

返回值:

```
lvector_double2 ret;
```

```
lvector double ret.x = {a1 的整数部分, a2 的整数部分};
```

```
lvector double ret.y = {a1 的小数部分, a2 的小数部分};
```

2.8.9 `fmax` 函数

1. `vm_fmaxd16`

接口:

```
lvector double  vm_fmaxd16(lvector double a,  lvector double b);
```

说明:

双精度向量求最大值函数, 该函数求取输入的 `lvector double` 向量类型变量 `a` 和 `b` 中的对应的各浮点数 `ai` 与 `bi` 的最大值 ($\max(ai, bi)$), 并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `fmax` 函数的向量版本。

举例：

输入：

```
lvector double a = {a1, ..., a15 };
```

```
lvector double b = {b1, ..., b15 };
```

返回值：

```
lvector double ret;
```

```
ret = {fmax(a1, b1), ..., fmax(a15, b15)};
```

2. vm_fmaxf32

接口：

```
lvector float vm_fmaxf32(lvector float a, lvector float b);
```

说明：

单精度向量求最大值函数，该函数求取输入的 lvector float 向量类型变量 a 和 b 中的对应的各浮点数 ai 与 bi 的最大值 ($\max(a_i, b_i)$)，并将结果存储于 lvector float 变量中返回。该函数是 math.h 中 fmaxf 函数的向量版本。

举例：

输入：

```
lvector float a = {a1, ..., a31};
```

```
lvector float b = {b1, ..., b31};
```

返回值：

```
lvector float ret;
```

```
ret = {fmaxf(a1, b1), ..., fmaxf(a31, b31)};
```

2.8.10 fmin 函数

1. vm_fmind16

接口：

```
lvector double vm_fmind16(lvector double a, lvector double b);
```

说明：

双精度向量求最小值函数，该函数求取输入的 lvector double 向量类型变量 a 和 b 中的对应的各浮点数 ai 与 bi 的最小值 ($\min(a_i, b_i)$)，并将结果分别

存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `fmin` 函数的向量版本。

举例：

输入：

```
lvector double a = {a1, ..., a15 };
```

```
lvector double b = {b1, ..., b15 };
```

返回值：

```
lvector double ret;
```

```
ret = {fmin(a1, b1), ..., fmin(a15, b15)};
```

2. `vm_fminf32`

接口：

```
lvector float vm_fminf32(lvector float a, lvector float b);
```

说明：

单精度向量求最小值函数，该函数求取输入的 `lvector float` 向量类型变量 `a` 和 `b` 中的对应的各浮点数 `ai` 与 `bi` 的最小值 ($\min(ai, bi)$)，并将结果分别存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `fminf` 函数的向量版本。

举例：

输入：

```
lvector float a = {a1, ..., a31};
```

```
lvector float b = {b1, ..., b31};
```

返回值：

```
lvector float ret;
```

```
ret = {fmaxf(a1, b1), ..., fmaxf(a31, b31)};
```

2.8.11 `fdim` 函数

1. `vm_fdimd16`

接口：

```
lvector double vm_fdimd16(lvector double a, lvector double b);
```

说明：

双精度向量求取正差值函数 (`fdim`)，该函数求取输入的 `lvector double` 向

量类型变量 a 和 b 中的对应的各浮点数 a_i 与 b_i 的差值 ($a_i - b_i$), 如果 $a_i < b_i$, 则为 0.0。并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `fdim` 函数的向量版本。

2. `vm_fdimf32`

接口:

```
lvector float vm_fdimf32(lvector float a, lvector float b);
```

说明:

单精度向量求取正差值函数 (`fdim`), 该函数求取输入的 `lvector float` 向量类型变量 a 和 b 中的对应的各浮点数 a_i 与 b_i 的差值 ($a_i - b_i$), 如果 $a_i < b_i$, 则为 0.0。并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `fdimf` 函数的向量版本。

2.8.12 `copysign` 函数

1. `vm_copysignd16`

接口:

```
lvector double vm_copysignd16(lvector double a, lvector double b);
```

说明:

双精度向量浮点数符号复制函数, 该函数针对输入的 `lvector double` 向量类型变量 a 和 b , 将 a 的各元素 a_i 的浮点数正/负符号复制给对应 b 的元素 b_i , 并将结果存储于 `lvector double` 变量中返回。该函数是 `math.h` 中 `copysign` 函数的向量版本。

2. `vm_copysignf32`

接口:

```
lvector float vm_copysignf32(lvector float a, lvector float b);
```

说明:

单精度向量浮点数符号复制函数, 该函数针对输入的 `lvector float` 向量类型变量 a 和 b , 将 a 的各元素 a_i 的浮点数正/负符号复制给对应 b 的元素 b_i , 并将结果存储于 `lvector float` 变量中返回。该函数是 `math.h` 中 `copysignf` 函数的向量版本。

2.8.13 nextafter 函数

1. vm_nextafterd16

接口:

```
lvector double vm_nextafterd16(lvector double a, lvector double b);
```

说明:

双精度向量浮点数 nextafter, 该函数针对输入的 lvector double 向量类型变量 a 和 b, 获取 a 中各元素 ai 在对应 b 的元素 bi 方向的最近的某个浮点数, 并将结果存储于 lvector double 变量中返回。该函数是 math.h 中 nextafter 函数的向量版本。

2. vm_nextafterf32

接口:

```
lvector float vm_nextafterf32(lvector float a, lvector float b);
```

说明:

单精度向量浮点数 nextafter, 该函数针对输入的 lvector float 向量类型变量 a 和 b, 获取 a 中各元素 ai 在对应 b 的元素 bi 方向的最近的某个浮点数, 并将结果存储于 lvector float 变量中返回。该函数是 math.h 中 nextafterf 函数的向量版本。

2.9 除法与求倒数函数

2.9.1 fdiv 函数

1. vm_fdivd16;

接口:

```
lvector double vm_fdivd16(lvector double a, lvector double b);
```

说明:

双精度向量浮点除法函数, 该函数计算 lvector double 向量类型变量 a 和 b 中的对应的每个浮点元素 ai 和 bi 的除法值(ai/bi), 并将结果存储于 lvector double 变量中返回。

精度情况:

该函数有极少数结果误差为 2.0ulp, 其余结果误差为 0ulp。

非规格数支持:

不支持非规格数运算。

2. **vm_fdivf32;**

接口:

```
lvector float vm_fdivf32(lvector float a, lvector float b);
```

说明:

单精度向量浮点除法函数, 该函数计算 lvector float 向量类型变量 a 和 b 中的对应的每个浮点元素 a_i 和 b_i 的除法值 (a_i/b_i), 并将结果存储于 lvector float 变量中返回。

精度情况:

该函数有极少数结果误差为 2.0ulp, 其余结果误差为 0ulp。

非规格数支持:

不支持非规格数运算。

2.9.2 **frec 函数**

1. **vm_frecd16;**

接口:

```
lvector double vm_frecd16(lvector double a);
```

说明:

双精度向量求倒数函数, 该函数计算 lvector double 向量类型变量 a 每个元素的倒数值, 并将结果存储于 lvector double 变量中返回。

精度情况:

该函数有极少数结果误差为 1.0ulp, 其余结果误差为 0ulp。

非规格数支持:

不支持非规格数运算。

2. **vm_frexf32;**

接口:

```
lvector float vm_frexf32(lvector float a);
```

说明:

单精度向量求倒数函数，该函数计算 `lvector float` 向量类型变量 `a` 每个元素的倒数值，并将结果存储于 `lvector float` 变量中返回。

精度情况：

该函数有极少数结果误差为 `1.0ulp`，其余结果误差为 `0ulp`。

非规格数支持：

不支持非规格数运算。