Vector Math Library
C Reference

Table of Contents

Structures (Array of Structures/AoS)

VmathMatrix3

A 3x3 matrix in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathMatrix3 {
        VmathVector3 col0;
        VmathVector3 col1;
        VmathVector3 col2;
} VmathMatrix3;
```

Members

col0	Column 0
col1	Column 1
col2	Column 2

Description

A struct representing a 3x3 matrix stored in array-of-structures (AoS) format

VmathMatrix4

A 4x4 matrix in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathMatrix4 {
        VmathVector4 col0;
        VmathVector4 col1;
        VmathVector4 col2;
        VmathVector4 col3;
} VmathMatrix4;
```

Members

col0	Column 0
col1	Column 1
col2	Column 2
col3	Column 3

Description

A struct representing a 4x4 matrix stored in array-of-structures (AoS) format

VmathPoint3

A 3-D point in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathPoint3 {
        vec_float4 vec128;
} VmathPoint3;
```

Members

vec128 Vector float data

Description

A struct representing a 3-D point stored in array-of-structures (AoS) format

VmathQuat

A quaternion in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathQuat {
        vec_float4 vec128;
} VmathQuat;
```

Members

vec128 Vector float data

Description

A struct representing a quaternion stored in array-of-structures (AoS) format

VmathTransform3

A 3x4 transformation matrix in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathTransform3 {
        VmathVector3 col0;
        VmathVector3 col1;
        VmathVector3 col2;
        VmathVector3 col3;
} VmathTransform3;
```

Members

col0	Column 0
col1	Column 1
col2	Column 2
col3	Column 3

Description

A struct representing a 3x4 transformation matrix stored in array-of-structures (AoS) format

VmathVector3

A 3-D vector in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathVector3 {
        vec_float4 vec128;
} VmathVector3;
```

Members

vec128 Vector float data

Description

A struct representing a 3-D vector stored in array-of-structures (AoS) format

VmathVector4

A 4-D vector in array-of-structures format.

Definition

```
#include <vectormath/c/vectormath_aos.h> or <vectormath/c/vectormath_aos_v.h>
typedef struct _VmathVector4 {
        vec_float4 vec128;
} VmathVector4;
```

Members

vec128 Vector float data

Description

A struct representing a 4-D vector stored in array-of-structures (AoS) format

Structures (Structure of Arrays/SoA)

VmathSoaMatrix3

A set of four 3x3 matrices in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaMatrix3 {
        VmathSoaVector3 col0;
        VmathSoaVector3 col1;
        VmathSoaVector3 col2;
} VmathSoaMatrix3;
```

Members

col0	Column 0 of four 3x3 matrices in SoA format
col1	Column 1 of four 3x3 matrices in SoA format
col2	Column 2 of four 3x3 matrices in SoA format

Description

A struct representing a set of four 3x3 matrices stored in structure-of-arrays (SoA) format

VmathSoaMatrix4

A set of four 4x4 matrices in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaMatrix4 {
        VmathSoaVector4 col0;
        VmathSoaVector4 col1;
        VmathSoaVector4 col2;
        VmathSoaVector4 col3;
} VmathSoaMatrix4;
```

Members

col0	Column 0 of four 4x4 matrices in SoA format
col1	Column 1 of four 4x4 matrices in SoA format
col2	Column 2 of four 4x4 matrices in SoA format
col3	Column 3 of four 4x4 matrices in SoA format

Description

A struct representing a set of four 4x4 matrices stored in structure-of-arrays (SoA) format

VmathSoaPoint3

A set of four 3-D points in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaPoint3 {
        vec_float4 x;
        vec_float4 y;
        vec_float4 z;
} VmathSoaPoint3;
```

Members

x	A set of four x elements in SoA format
y	A set of four y elements in SoA format
z	A set of four z elements in SoA format

Description

A struct representing a set of four 3-D points stored in structure-of-arrays (SoA) format

VmathSoaQuat

A set of four quaternions in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaQuat {
        vec_float4 x;
        vec_float4 y;
        vec_float4 z;
        vec_float4 w;
} VmathSoaQuat;
```

Members

x	A set of four x elements in SoA format
y	A set of four y elements in SoA format
z	A set of four z elements in SoA format
w	A set of four w elements in SoA format

Description

A struct representing a set of four quaternions stored in structure-of-arrays (SoA) format

VmathSoaTransform3

A set of four 3x4 transformation matrices in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaTransform3 {
        VmathSoaVector3 col0;
        VmathSoaVector3 col1;
        VmathSoaVector3 col2;
        VmathSoaVector3 col3;
} VmathSoaTransform3;
```

Members

col0	Column 0 of four 3x4 transformation matrices in SoA format
col1	Column 1 of four 3x4 transformation matrices in SoA format
col2	Column 2 of four 3x4 transformation matrices in SoA format
col3	Column 3 of four 3x4 transformation matrices in SoA format

Description

A struct representing a set of four 3x4 transformation matrices stored in structure-of-arrays (SoA) format

VmathSoaVector3

A set of four 3-D vectors in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaVector3 {
        vec_float4 x;
        vec_float4 y;
        vec_float4 z;
} VmathSoaVector3;
```

Members

x	A set of four x elements in SoA format
y	A set of four y elements in SoA format
z	A set of four z elements in SoA format

Description

A struct representing a set of four 3-D vectors stored in structure-of-arrays (SoA) format

VmathSoaVector4

A set of four 4-D vectors in structure-of-arrays format.

Definition

```
#include <vectormath/c/vectormath_soa.h> or <vectormath/c/vectormath_soa_v.h>
typedef struct _VmathSoaVector4 {
        vec_float4 x;
        vec_float4 y;
        vec_float4 z;
        vec_float4 w;
} VmathSoaVector4;
```

Members

x	A set of four x elements in SoA format
y	A set of four y elements in SoA format
z	A set of four z elements in SoA format
w	A set of four w elements in SoA format

Description

A struct representing a set of four 4-D vectors stored in structure-of-arrays (SoA) format

3-D Vector Functions (AoS, by reference)

vmathV3AbsPerElem

Compute the absolute value of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3AbsPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the absolute value of the corresponding element of vec
vec	3-D vector

Return Values

None

Description

Compute the absolute value of each element of a 3-D vector.

vmathV3Add

Add two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Add(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	Sum of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Add two 3-D vectors.

vmathV3AddP3

Add a 3-D vector to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3AddP3(
        VmathPoint3 *result,
        const VmathVector3 *vec,
        const VmathPoint3 *pnt
);
```

Arguments

result	Sum of the specified 3-D vector and 3-D point
vec	3-D vector
pnt	3-D point

Return Values

None

Description

Add a 3-D vector to a 3-D point.

vmathV3Copy

Copy a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Copy(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	The constructed result
vec	3-D vector

Return Values

None

Description

Construct a copy of a 3-D vector.

vmathV3CopySignPerElem

Copy sign from one 3-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3CopySignPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	3-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathV3Cross

Compute cross product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Cross(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	Cross product of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Compute cross product of two 3-D vectors.

vmathV3CrossMatrix

Cross-product matrix of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3CrossMatrix(
        VmathMatrix3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	Cross-product matrix of *vec*
vec	3-D vector

Return Values

None

Description

Compute a matrix that, when multiplied by a 3-D vector, produces the same result as a cross product with that 3-D vector.

vmathV3CrossMatrixMul

Create cross-product matrix and multiply.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3CrossMatrixMul(
        VmathMatrix3 *result,
        const VmathVector3 *vec,
        const VmathMatrix3 *mat
);
```

Arguments

result	Product of cross-product matrix of *vec* and *mat*
vec	3-D vector
mat	3x3 matrix

Return Values

None

Description

Multiply a cross-product matrix by another matrix.

Notes

Faster than separately creating a cross-product matrix and multiplying.

vmathV3DivPerElem

Divide two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3DivPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the quotient of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Divide two 3-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathV3Dot

Compute the dot product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3Dot(
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Dot product of the specified 3-D vectors

Description

Compute the dot product of two 3-D vectors.

vmathV3Get128

Get vector float data from a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline vec_float4 vmathV3Get128(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Internal vector float data

Description

Get internal vector float data from a 3-D vector.

vmathV3GetElem

Get an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3GetElem(
        const VmathVector3 *vec,
        int idx
);
```

Arguments

vec	3-D vector
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathV3GetX

Get the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3GetX(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

x element of a 3-D vector

Description

Get the x element of a 3-D vector.

vmathV3GetY

Get the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3GetY(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

y element of a 3-D vector

Description

Get the y element of a 3-D vector.

vmathV3GetZ

Get the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3GetZ(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

z element of a 3-D vector

Description

Get the z element of a 3-D vector.

vmathV3Length

Compute the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3Length(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Length of the specified 3-D vector

Description

Compute the length of a 3-D vector.

vmathV3LengthSqr

Compute the square of the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3LengthSqr(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Square of the length of the specified 3-D vector

Description

Compute the square of the length of a 3-D vector.

vmathV3Lerp

Linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Lerp(
        VmathVector3 *result,
        float t,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	Interpolated 3-D vector
t	Interpolation parameter
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Linearly interpolate between two 3-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathV3LoadXYZArray

Load four three-float 3-D vectors, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3LoadXYZArray(
        VmathVector3 *vec0,
        VmathVector3 *vec1,
        VmathVector3 *vec2,
        VmathVector3 *vec3,
        const vec_float4 *threeQuads
);
```

Arguments

vec0	An output 3-D vector
vec1	An output 3-D vector
vec2	An output 3-D vector
vec3	An output 3-D vector
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D vectors, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four 3-D vectors.

vmathV3MakeFrom128

Set vector float data in a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeFrom128(
        VmathVector3 *result,
        vec_float4 vf4
);
```

Arguments

result	The constructed 3-D vector
vf4	Scalar value

Return Values

None

Description

Construct a 3-D vector whose internal vector float data is set to the vector float argument.

vmathV3MakeFromElems

Construct a 3-D vector from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeFromElems(
        VmathVector3 *result,
        float x,
        float y,
        float z
);
```

Arguments

result	The 3-D vector that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value

Return Values

None

Description

Construct a 3-D vector containing the specified x, y, and z elements.

vmathV3MakeFromP3

Copy elements from a 3-D point into a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeFromP3(
        VmathVector3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	The constructed 3-D vector
pnt	3-D point

Return Values

None

Description

Construct a 3-D vector containing the x, y, and z elements of the specified 3-D point.

vmathV3MakeFromScalar

Set all elements of a 3-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeFromScalar(
        VmathVector3 *result,
        float scalar
);
```

Arguments

result	The constructed 3-D vector
scalar	Scalar value

Return Values

None

Description

Construct a 3-D vector with all elements set to the scalar value argument.

vmathV3MakeXAxis

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeXAxis(
        VmathVector3 *result
);
```

Arguments

result The constructed 3-D vector

Return Values

None

Description

Construct a 3-D vector equal to (1,0,0).

vmathV3MakeYAxis

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeYAxis(
        VmathVector3 *result
);
```

Arguments

result The constructed 3-D vector

Return Values

None

Description

Construct a 3-D vector equal to (0,1,0).

vmathV3MakeZAxis

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MakeZAxis(
        VmathVector3 *result
);
```

Arguments

result The constructed 3-D vector

Return Values

None

Description

Construct a 3-D vector equal to (0,0,1).

vmathV3MaxElem

Maximum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3MaxElem(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 3-D vector.

vmathV3MaxPerElem

Maximum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MaxPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors.

vmathV3MinElem

Minimum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3MinElem(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 3-D vector.

vmathV3MinPerElem

Minimum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MinPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the minimum of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the minimum of the corresponding elements of two specified 3-D vectors.

vmathV3MulPerElem

Multiply two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3MulPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the product of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Multiply two 3-D vectors element by element.

vmathV3Neg

Negate all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Neg(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	3-D vector containing negated elements of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Negate all elements of a 3-D vector.

vmathV3Normalize

Normalize a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Normalize(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	The specified 3-D vector scaled to unit length
vec	3-D vector

Return Values

None

Description

Compute a normalized 3-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathV3Outer

Outer product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Outer(
        VmathMatrix3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	The 3x3 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Compute the outer product of two 3-D vectors.

vmathV3Print

Print a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Print(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

None

Description

Print a 3-D vector. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV3Prints

Print a 3-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Prints(
        const VmathVector3 *vec,
        const char *name
);
```

Arguments

vec	3-D vector
name	String printed with the 3-D vector

Return Values

None

Description

Print a 3-D vector and an associated string identifier. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV3RecipPerElem

Compute the reciprocal of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3RecipPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathV3RowMul

Pre-multiply a row vector by a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3RowMul(
        VmathVector3 *result,
        const VmathVector3 *vec,
        const VmathMatrix3 *mat
);
```

Arguments

result	Product of a row-vector and a 3x3 matrix
vec	3-D vector
mat	3x3 matrix

Return Values

None

Description

Transpose a 3-D vector into a row vector and pre-multiply by 3x3 matrix.

Notes

Slower than column post-multiply.

vmathV3RsqrtPerElem

Compute the reciprocal square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3RsqrtPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathV3ScalarDiv

Divide a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3ScalarDiv(
        VmathVector3 *result,
        const VmathVector3 *vec,
        float scalar
);
```

Arguments

result	Quotient of the specified 3-D vector and scalar
vec	3-D vector
scalar	Scalar value

Return Values

None

Description

Divide a 3-D vector by a scalar.

vmathV3ScalarMul

Multiply a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3ScalarMul(
        VmathVector3 *result,
        const VmathVector3 *vec,
        float scalar
);
```

Arguments

result	Product of the specified 3-D vector and scalar
vec	3-D vector
scalar	Scalar value

Return Values

None

Description

Multiply a 3-D vector by a scalar.

vmathV3Select

Conditionally select between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Select(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1,
        unsigned int select1
);
```

Arguments

result	Equal to `vec0` if `select1` == 0, or to `vec1` if `select1` != 0
vec0	3-D vector
vec1	3-D vector
select1	False selects the vec0 argument, true selects the vec1 argument

Return Values

None

Description

Conditionally select one of the 3-D vector arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of `select1` to a VMX register may use more processing time than a branch.

vmathV3SetElem

Set an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3SetElem(
        VmathVector3 *result,
        int idx,
        float value
);
```

Arguments

result	An output 3-D vector
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathV3SetX

Set the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3SetX(
        VmathVector3 *result,
        float x
);
```

Arguments

result	An output 3-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D vector to the specified scalar value.

vmathV3SetY

Set the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3SetY(
        VmathVector3 *result,
        float y
);
```

Arguments

result	An output 3-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D vector to the specified scalar value.

vmathV3SetZ

Set the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3SetZ(
        VmathVector3 *result,
        float z
);
```

Arguments

result An output 3-D vector
z Scalar value

Return Values

None

Description

Set the z element of a 3-D vector to the specified scalar value.

vmathV3Slerp

Spherical linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Slerp(
        VmathVector3 *result,
        float t,
        const VmathVector3 *unitVec0,
        const VmathVector3 *unitVec1
);
```

Arguments

result	Interpolated 3-D vector
t	Interpolation parameter
unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

None

Description

Perform spherical linear interpolation between two 3-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp `t` between 0 and 1.

vmathV3SqrtPerElem

Compute the square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3SqrtPerElem(
        VmathVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathV3StoreHalfFloats

Store eight 3-D vectors as half-floats.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3StoreHalfFloats(
        const VmathVector3 *vec0,
        const VmathVector3 *vec1,
        const VmathVector3 *vec2,
        const VmathVector3 *vec3,
        const VmathVector3 *vec4,
        const VmathVector3 *vec5,
        const VmathVector3 *vec6,
        const VmathVector3 *vec7,
        vec_ushort8 *threeQuads
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
vec2	3-D vector
vec3	3-D vector
vec4	3-D vector
vec5	3-D vector
vec6	3-D vector
vec7	3-D vector
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight 3-D vectors in three quadwords of half-float values. The output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathV3StoreXYZ

Store x, y, and z elements of a 3-D vector in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3StoreXYZ(
        const VmathVector3 *vec,
        vec_float4 *quad
);
```

Arguments

vec	3-D vector
quad	Pointer to a quadword in which x, y, and z will be stored

Return Values

None

Description

Store x, y, and z elements of a 3-D vector in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

vmathV3StoreXYZArray

Store four 3-D vectors in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3StoreXYZArray(
        const VmathVector3 *vec0,
        const VmathVector3 *vec1,
        const VmathVector3 *vec2,
        const VmathVector3 *vec3,
        vec_float4 *threeQuads
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
vec2	3-D vector
vec3	3-D vector
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four 3-D vectors in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathV3Sub

Subtract a 3-D vector from another 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV3Sub(
        VmathVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1
);
```

Arguments

result	Difference of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Subtract a 3-D vector from another 3-D vector.

vmathV3Sum

Compute the sum of all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV3Sum(
        const VmathVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 3-D vector.

4-D Vector Functions (AoS, by reference)

vmathV4AbsPerElem

Compute the absolute value of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4AbsPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the absolute value of the corresponding element of vec
vec	4-D vector

Return Values

None

Description

Compute the absolute value of each element of a 4-D vector.

vmathV4Add

Add two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Add(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	Sum of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Add two 4-D vectors.

vmathV4Copy

Copy a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Copy(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	The constructed result
vec	4-D vector

Return Values

None

Description

Construct a copy of a 4-D vector.

vmathV4CopySignPerElem

Copy sign from one 4-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4CopySignPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	4-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathV4DivPerElem

Divide two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4DivPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the quotient of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Divide two 4-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathV4Dot

Compute the dot product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4Dot(
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

vec0 4-D vector
vec1 4-D vector

Return Values

Dot product of the specified 4-D vectors

Description

Compute the dot product of two 4-D vectors.

vmathV4Get128

Get vector float data from a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline vec_float4 vmathV4Get128(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Internal vector float data

Description

Get internal vector float data from a 4-D vector.

vmathV4GetElem

Get an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4GetElem(
        const VmathVector4 *vec,
        int idx
);
```

Arguments

vec	4-D vector
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathV4GetW

Get the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4GetW(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

w element of a 4-D vector

Description

Get the w element of a 4-D vector.

vmathV4GetX

Get the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4GetX(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

x element of a 4-D vector

Description

Get the x element of a 4-D vector.

vmathV4GetXYZ

Get the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4GetXYZ(
        VmathVector3 *result,
        const VmathVector4 *vec
);
```

Arguments

result	3-D vector containing x, y, and z elements
vec	4-D vector

Return Values

None

Description

Extract a 4-D vector's x, y, and z elements into a 3-D vector.

vmathV4GetY

Get the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4GetY(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

y element of a 4-D vector

Description

Get the y element of a 4-D vector.

vmathV4GetZ

Get the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4GetZ(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

z element of a 4-D vector

Description

Get the z element of a 4-D vector.

vmathV4Length

Compute the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4Length(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Length of the specified 4-D vector

Description

Compute the length of a 4-D vector.

vmathV4LengthSqr

Compute the square of the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4LengthSqr(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Square of the length of the specified 4-D vector

Description

Compute the square of the length of a 4-D vector.

vmathV4Lerp

Linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Lerp(
        VmathVector4 *result,
        float t,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	Interpolated 4-D vector
t	Interpolation parameter
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Linearly interpolate between two 4-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathV4MakeFrom128

Set vector float data in a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFrom128(
        VmathVector4 *result,
        vec_float4 vf4
);
```

Arguments

result	The constructed 4-D vector
vf4	Scalar value

Return Values

None

Description

Construct a 4-D vector whose internal vector float data is set to the vector float argument.

vmathV4MakeFromElems

Construct a 4-D vector from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFromElems(
        VmathVector4 *result,
        float x,
        float y,
        float z,
        float w
);
```

Arguments

result	The 4-D vector that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value
w	Scalar value

Return Values

None

Description

Construct a 4-D vector containing the specified x, y, z, and w elements.

vmathV4MakeFromP3

Copy x, y, and z from a 3-D point into a 4-D vector, and set w to 1.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFromP3(
        VmathVector4 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	The constructed 4-D vector
pnt	3-D point

Return Values

None

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D point and with the w element set to 1.

vmathV4MakeFromQ

Copy elements from a quaternion into a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFromQ(
        VmathVector4 *result,
        const VmathQuat *quat
);
```

Arguments

result	The constructed 4-D vector
quat	Quaternion

Return Values

None

Description

Construct a 4-D vector containing the x, y, z, and w elements of the specified quaternion.

vmathV4MakeFromScalar

Set all elements of a 4-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFromScalar(
        VmathVector4 *result,
        float scalar
);
```

Arguments

result	The constructed 4-D vector
scalar	Scalar value

Return Values

None

Description

Construct a 4-D vector with all elements set to the scalar value argument.

vmathV4MakeFromV3

Copy x, y, and z from a 3-D vector into a 4-D vector, and set w to 0.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFromV3(
        VmathVector4 *result,
        const VmathVector3 *vec
);
```

Arguments

result	The constructed 4-D vector
vec	3-D vector

Return Values

None

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to 0.

vmathV4MakeFromV3Scalar

Construct a 4-D vector from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeFromV3Scalar(
        VmathVector4 *result,
        const VmathVector3 *xyz,
        float w
);
```

Arguments

result	The constructed result
xyz	3-D vector
w	Scalar value

Return Values

None

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathV4MakeWAxis

Construct w axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeWAxis(
        VmathVector4 *result
);
```

Arguments

result The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (0,0,0,1).

vmathV4MakeXAxis

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeXAxis(
        VmathVector4 *result
);
```

Arguments

result The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (1,0,0,0).

vmathV4MakeYAxis

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeYAxis(
        VmathVector4 *result
);
```

Arguments

result The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (0,1,0,0).

vmathV4MakeZAxis

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MakeZAxis(
        VmathVector4 *result
);
```

Arguments

result The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (0,0,1,0).

vmathV4MaxElem

Maximum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4MaxElem(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 4-D vector.

vmathV4MaxPerElem

Maximum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MaxPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors.

vmathV4MinElem

Minimum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4MinElem(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 4-D vector.

vmathV4MinPerElem

Minimum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MinPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the minimum of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the minimum of the corresponding elements of two specified 4-D vectors.

vmathV4MulPerElem

Multiply two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4MulPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the product of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Multiply two 4-D vectors element by element.

vmathV4Neg

Negate all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Neg(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	4-D vector containing negated elements of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Negate all elements of a 4-D vector.

vmathV4Normalize

Normalize a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Normalize(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	The specified 4-D vector scaled to unit length
vec	4-D vector

Return Values

None

Description

Compute a normalized 4-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathV4Outer

Outer product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Outer(
        VmathMatrix4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	The 4x4 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Compute the outer product of two 4-D vectors.

vmathV4Print

Print a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Print(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

None

Description

Print a 4-D vector. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV4Prints

Print a 4-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Prints(
        const VmathVector4 *vec,
        const char *name
);
```

Arguments

vec	4-D vector
name	String printed with the 4-D vector

Return Values

None

Description

Print a 4-D vector and an associated string identifier. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV4RecipPerElem

Compute the reciprocal of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4RecipPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathV4RsqrtPerElem

Compute the reciprocal square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4RsqrtPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathV4ScalarDiv

Divide a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4ScalarDiv(
        VmathVector4 *result,
        const VmathVector4 *vec,
        float scalar
);
```

Arguments

result	Quotient of the specified 4-D vector and scalar
vec	4-D vector
scalar	Scalar value

Return Values

None

Description

Divide a 4-D vector by a scalar.

vmathV4ScalarMul

Multiply a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4ScalarMul(
        VmathVector4 *result,
        const VmathVector4 *vec,
        float scalar
);
```

Arguments

result	Product of the specified 4-D vector and scalar
vec	4-D vector
scalar	Scalar value

Return Values

None

Description

Multiply a 4-D vector by a scalar.

vmathV4Select

Conditionally select between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Select(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1,
        unsigned int select1
);
```

Arguments

result	Equal to *vec0* if *select1* == 0, or to *vec1* if *select1* != 0
vec0	4-D vector
vec1	4-D vector
select1	False selects the vec0 argument, true selects the vec1 argument

Return Values

None

Description

Conditionally select one of the 4-D vector arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathV4SetElem

Set an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SetElem(
        VmathVector4 *result,
        int idx,
        float value
);
```

Arguments

result	An output 4-D vector
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathV4SetW

Set the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SetW(
        VmathVector4 *result,
        float w
);
```

Arguments

result	An output 4-D vector
w	Scalar value

Return Values

None

Description

Set the w element of a 4-D vector to the specified scalar value.

vmathV4SetX

Set the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SetX(
        VmathVector4 *result,
        float x
);
```

Arguments

result	An output 4-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 4-D vector to the specified scalar value.

vmathV4SetXYZ

Set the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SetXYZ(
        VmathVector4 *result,
        const VmathVector3 *vec
);
```

Arguments

result	An output 4-D vector
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathV4SetY

Set the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SetY(
        VmathVector4 *result,
        float y
);
```

Arguments

result	An output 4-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 4-D vector to the specified scalar value.

vmathV4SetZ

Set the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SetZ(
        VmathVector4 *result,
        float z
);
```

Arguments

result	An output 4-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 4-D vector to the specified scalar value.

vmathV4Slerp

Spherical linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Slerp(
        VmathVector4 *result,
        float t,
        const VmathVector4 *unitVec0,
        const VmathVector4 *unitVec1
);
```

Arguments

result	Interpolated 4-D vector
t	Interpolation parameter
unitVec0	4-D vector, expected to be unit-length
unitVec1	4-D vector, expected to be unit-length

Return Values

None

Description

Perform spherical linear interpolation between two 4-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp *t* between 0 and 1.

vmathV4SqrtPerElem

Compute the square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4SqrtPerElem(
        VmathVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathV4StoreHalfFloats

Store four 4-D vectors as half-floats.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4StoreHalfFloats(
        const VmathVector4 *vec0,
        const VmathVector4 *vec1,
        const VmathVector4 *vec2,
        const VmathVector4 *vec3,
        vec_ushort8 *twoQuads
);
```

Arguments

vec0	4-D vector
vec1	4-D vector
vec2	4-D vector
vec3	4-D vector
twoQuads	An output array of 2 quadwords containing 16 half-floats

Return Values

None

Description

Store four 4-D vectors in two quadwords of half-float values. The output is {x0,y0,z0,w0,x1,y1,z1,w1,x2,y2,z2,w2,x3,y3,z3,w3}.

vmathV4Sub

Subtract a 4-D vector from another 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathV4Sub(
        VmathVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1
);
```

Arguments

result	Difference of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Subtract a 4-D vector from another 4-D vector.

vmathV4Sum

Compute the sum of all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathV4Sum(
        const VmathVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 4-D vector.

Point Functions (AoS, by reference)

vmathP3AbsPerElem

Compute the absolute value of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3AbsPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the absolute value of the corresponding element of pnt
pnt	3-D point

Return Values

None

Description

Compute the absolute value of each element of a 3-D point.

vmathP3AddV3

Add a 3-D point to a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3AddV3(
        VmathPoint3 *result,
        const VmathPoint3 *pnt,
        const VmathVector3 *vec
);
```

Arguments

result	Sum of the specified 3-D point and 3-D vector
pnt	3-D point
vec	3-D vector

Return Values

None

Description

Add a 3-D point to a 3-D vector.

vmathP3Copy

Copy a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Copy(
        VmathPoint3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	The constructed result
pnt	3-D point

Return Values

None

Description

Construct a copy of a 3-D point.

vmathP3CopySignPerElem

Copy sign from one 3-D point to another, per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3CopySignPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element has the magnitude of the corresponding element of *pnt0* and the sign of the corresponding element of *pnt1*
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

For each element, create a value composed of the magnitude of *pnt0* and the sign of *pnt1*.

vmathP3Dist

Compute the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3Dist(
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Distance between two 3-D points

Description

Compute the distance between two 3-D points.

vmathP3DistFromOrigin

Compute the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3DistFromOrigin(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Distance of a 3-D point from the origin

Description

Compute the distance of a 3-D point from the coordinate-system origin.

vmathP3DistSqr

Compute the square of the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3DistSqr(
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

pnt0 3-D point
pnt1 3-D point

Return Values

Square of the distance between two 3-D points

Description

Compute the square of the distance between two 3-D points.

vmathP3DistSqrFromOrigin

Compute the square of the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3DistSqrFromOrigin(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Square of the distance of a 3-D point from the origin

Description

Compute the square of the distance of a 3-D point from the coordinate-system origin.

vmathP3DivPerElem

Divide two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3DivPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the quotient of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Divide two 3-D points element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathP3Get128

Get vector float data from a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline vec_float4 vmathP3Get128(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Internal vector float data

Description

Get internal vector float data from a 3-D point.

vmathP3GetElem

Get an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3GetElem(
        const VmathPoint3 *pnt,
        int idx
);
```

Arguments

pnt	3-D point
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathP3GetX

Get the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3GetX(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

x element of a 3-D point

Description

Get the x element of a 3-D point.

vmathP3GetY

Get the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3GetY(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

y element of a 3-D point

Description

Get the y element of a 3-D point.

vmathP3GetZ

Get the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3GetZ(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

z element of a 3-D point

Description

Get the z element of a 3-D point.

vmathP3Lerp

Linear interpolation between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Lerp(
        VmathPoint3 *result,
        float t,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	Interpolated 3-D point
t	Interpolation parameter
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Linearly interpolate between two 3-D points.

Notes

Does not clamp *t* between 0 and 1.

vmathP3LoadXYZArray

Load four three-float 3-D points, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3LoadXYZArray(
        VmathPoint3 *pnt0,
        VmathPoint3 *pnt1,
        VmathPoint3 *pnt2,
        VmathPoint3 *pnt3,
        const vec_float4 *threeQuads
);
```

Arguments

pnt0	An output 3-D point
pnt1	An output 3-D point
pnt2	An output 3-D point
pnt3	An output 3-D point
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D points, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four 3-D points.

vmathP3MakeFrom128

Set vector float data in a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MakeFrom128(
        VmathPoint3 *result,
        vec_float4 vf4
);
```

Arguments

result	The constructed 3-D point
vf4	Scalar value

Return Values

None

Description

Construct a 3-D point whose internal vector float data is set to the vector float argument.

vmathP3MakeFromElems

Construct a 3-D point from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MakeFromElems(
        VmathPoint3 *result,
        float x,
        float y,
        float z
);
```

Arguments

result	The 3-D point that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value

Return Values

None

Description

Construct a 3-D point containing the specified x, y, and z elements.

vmathP3MakeFromScalar

Set all elements of a 3-D point to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MakeFromScalar(
        VmathPoint3 *result,
        float scalar
);
```

Arguments

result	The constructed 3-D point
scalar	Scalar value

Return Values

None

Description

Construct a 3-D point with all elements set to the scalar value argument.

vmathP3MakeFromV3

Copy elements from a 3-D vector into a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MakeFromV3(
        VmathPoint3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	The constructed 3-D point
vec	3-D vector

Return Values

None

Description

Construct a 3-D point containing the x, y, and z elements of the specified 3-D vector.

vmathP3MaxElem

Maximum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3MaxElem(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Maximum value of all elements of *pnt*

Description

Compute the maximum value of all elements of a 3-D point.

vmathP3MaxPerElem

Maximum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MaxPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points.

vmathP3MinElem

Minimum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3MinElem(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Minimum value of all elements of *pnt*

Description

Compute the minimum value of all elements of a 3-D point.

vmathP3MinPerElem

Minimum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MinPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the minimum of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the minimum of the corresponding elements of two specified 3-D points.

vmathP3MulPerElem

Multiply two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3MulPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the product of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Multiply two 3-D points element by element.

vmathP3NonUniformScale

Apply non-uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3NonUniformScale(
        VmathPoint3 *result,
        const VmathPoint3 *pnt,
        const VmathVector3 *scaleVec
);
```

Arguments

result	3-D point in which each element is the product of the corresponding elements of the specified 3-D point and 3-D vector
pnt	3-D point
scaleVec	3-D vector

Return Values

None

Description

Apply non-uniform scale to a 3-D point.

vmathP3Print

Print a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Print(
        const VmathPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

None

Description

Print a 3-D point. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathP3Prints

Print a 3-D point and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Prints(
        const VmathPoint3 *pnt,
        const char *name
);
```

Arguments

pnt	3-D point
name	String printed with the 3-D point

Return Values

None

Description

Print a 3-D point and an associated string identifier. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathP3Projection

Scalar projection of a 3-D point on a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3Projection(
        const VmathPoint3 *pnt,
        const VmathVector3 *unitVec
);
```

Arguments

pnt	3-D point
unitVec	3-D vector, expected to be unit-length

Return Values

Scalar projection of the 3-D point on the unit-length 3-D vector

Description

Scalar projection of a 3-D point on a unit-length 3-D vector (dot product).

vmathP3RecipPerElem

Compute the reciprocal of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3RecipPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point
pnt	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function recipf4.

vmathP3RsqrtPerElem

Compute the reciprocal square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3RsqrtPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point
pnt	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathP3Scale

Apply uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Scale(
        VmathPoint3 *result,
        const VmathPoint3 *pnt,
        float scaleVal
);
```

Arguments

result	3-D point in which every element is multiplied by the scalar value
pnt	3-D point
scaleVal	Scalar value

Return Values

None

Description

Apply uniform scale to a 3-D point.

vmathP3Select

Conditionally select between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Select(
        VmathPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1,
        unsigned int select1
);
```

Arguments

result	Equal to *pnt0* if *select1* == 0, or to *pnt1* if *select1* != 0
pnt0	3-D point
pnt1	3-D point
select1	False selects the pnt0 argument, true selects the pnt1 argument

Return Values

None

Description

Conditionally select one of the 3-D point arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathP3SetElem

Set an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3SetElem(
        VmathPoint3 *result,
        int idx,
        float value
);
```

Arguments

result	An output 3-D point
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathP3SetX

Set the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3SetX(
        VmathPoint3 *result,
        float x
);
```

Arguments

result	An output 3-D point
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D point to the specified scalar value.

vmathP3SetY

Set the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3SetY(
        VmathPoint3 *result,
        float y
);
```

Arguments

result	An output 3-D point
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D point to the specified scalar value.

vmathP3SetZ

Set the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3SetZ(
        VmathPoint3 *result,
        float z
);
```

Arguments

result	An output 3-D point
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D point to the specified scalar value.

vmathP3SqrtPerElem

Compute the square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3SqrtPerElem(
        VmathPoint3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the square root of the corresponding element of the specified 3-D point
pnt	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathP3StoreHalfFloats

Store eight 3-D points as half-floats.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3StoreHalfFloats(
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1,
        const VmathPoint3 *pnt2,
        const VmathPoint3 *pnt3,
        const VmathPoint3 *pnt4,
        const VmathPoint3 *pnt5,
        const VmathPoint3 *pnt6,
        const VmathPoint3 *pnt7,
        vec_ushort8 *threeQuads
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
pnt2	3-D point
pnt3	3-D point
pnt4	3-D point
pnt5	3-D point
pnt6	3-D point
pnt7	3-D point
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight 3-D points in three quadwords of half-float values. The output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathP3StoreXYZ

Store x, y, and z elements of a 3-D point in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3StoreXYZ(
        const VmathPoint3 *pnt,
        vec_float4 *quad
);
```

Arguments

pnt	3-D point
quad	Pointer to a quadword in which x, y, and z will be stored

Return Values

None

Description

Store x, y, and z elements of a 3-D point in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

vmathP3StoreXYZArray

Store four 3-D points in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3StoreXYZArray(
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1,
        const VmathPoint3 *pnt2,
        const VmathPoint3 *pnt3,
        vec_float4 *threeQuads
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
pnt2	3-D point
pnt3	3-D point
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four 3-D points in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathP3Sub

Subtract a 3-D point from another 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3Sub(
        VmathVector3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1
);
```

Arguments

result	Difference of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Subtract a 3-D point from another 3-D point.

vmathP3SubV3

Subtract a 3-D vector from a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathP3SubV3(
        VmathPoint3 *result,
        const VmathPoint3 *pnt,
        const VmathVector3 *vec
);
```

Arguments

result	Difference of the specified 3-D point and 3-D vector
pnt	3-D point
vec	3-D vector

Return Values

None

Description

Subtract a 3-D vector from a 3-D point.

vmathP3Sum

Compute the sum of all elements of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathP3Sum(
        const VmathPoint3 *pnt
);
```

Arguments

`pnt` 3-D point

Return Values

Sum of all elements of `pnt`

Description

Compute the sum of all elements of a 3-D point.

Quaternion Functions (AoS, by reference)

vmathQAdd

Add two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQAdd(
        VmathQuat *result,
        const VmathQuat *quat0,
        const VmathQuat *quat1
);
```

Arguments

result	Sum of the specified quaternions
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Add two quaternions.

vmathQConj

Compute the conjugate of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQConj(
        VmathQuat *result,
        const VmathQuat *quat
);
```

Arguments

result	Conjugate of the specified quaternion
quat	Quaternion

Return Values

None

Description

Compute the conjugate of a quaternion.

vmathQCopy

Copy a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQCopy(
        VmathQuat *result,
        const VmathQuat *quat
);
```

Arguments

result	The constructed result
quat	Quaternion

Return Values

None

Description

Construct a copy of a quaternion.

vmathQDot

Compute the dot product of two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQDot(
        const VmathQuat *quat0,
        const VmathQuat *quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Dot product of the specified quaternions

Description

Compute the dot product of two quaternions.

vmathQGet128

Get vector float data from a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline vec_float4 vmathQGet128(
        const VmathQuat *quat
);
```

Arguments

quat Quaternion

Return Values

Internal vector float data

Description

Get internal vector float data from a quaternion.

vmathQGetElem

Get an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQGetElem(
        const VmathQuat *quat,
        int idx
);
```

Arguments

quat	Quaternion
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathQGetW

Get the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQGetW(
        const VmathQuat *quat
);
```

Arguments

quat Quaternion

Return Values

w element of a quaternion

Description

Get the w element of a quaternion.

vmathQGetX

Get the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQGetX(
        const VmathQuat *quat
);
```

Arguments

quat Quaternion

Return Values

x element of a quaternion

Description

Get the x element of a quaternion.

vmathQGetXYZ

Get the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQGetXYZ(
        VmathVector3 *result,
        const VmathQuat *quat
);
```

Arguments

result	3-D vector containing x, y, and z elements
quat	Quaternion

Return Values

None

Description

Extract a quaternion's x, y, and z elements into a 3-D vector.

vmathQGetY

Get the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQGetY(
        const VmathQuat *quat
);
```

Arguments

quat Quaternion

Return Values

y element of a quaternion

Description

Get the y element of a quaternion.

vmathQGetZ

Get the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQGetZ(
        const VmathQuat *quat
);
```

Arguments

quat Quaternion

Return Values

z element of a quaternion

Description

Get the z element of a quaternion.

vmathQLength

Compute the length of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQLength(
        const VmathQuat *quat
);
```

Arguments

quat Quaternion

Return Values

Length of the specified quaternion

Description

Compute the length of a quaternion.

vmathQLerp

Linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQLerp(
        VmathQuat *result,
        float t,
        const VmathQuat *quat0,
        const VmathQuat *quat1
);
```

Arguments

result	Interpolated quaternion
t	Interpolation parameter
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Linearly interpolate between two quaternions.

Notes

Does not clamp *t* between 0 and 1.

vmathQMakeFrom128

Set vector float data in a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeFrom128(
        VmathQuat *result,
        vec_float4 vf4
);
```

Arguments

result The constructed quaternion
vf4 Scalar value

Return Values

None

Description

Construct a quaternion whose internal vector float data is set to the vector float argument.

vmathQMakeFromElems

Construct a quaternion from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeFromElems(
        VmathQuat *result,
        float x,
        float y,
        float z,
        float w
);
```

Arguments

result	The quaternion that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value
w	Scalar value

Return Values

None

Description

Construct a quaternion containing the specified x, y, z, and w elements.

vmathQMakeFromM3

Convert a rotation matrix to a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeFromM3(
        VmathQuat *result,
        const VmathMatrix3 *rotMat
);
```

Arguments

result	The constructed result
rotMat	3x3 matrix, expected to be a rotation matrix

Return Values

None

Description

Construct a unit-length quaternion representing the same transformation as a rotation matrix.

vmathQMakeFromScalar

Set all elements of a quaternion to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeFromScalar(
        VmathQuat *result,
        float scalar
);
```

Arguments

result	The constructed quaternion
scalar	Scalar value

Return Values

None

Description

Construct a quaternion with all elements set to the scalar value argument.

vmathQMakeFromV3Scalar

Construct a quaternion from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeFromV3Scalar(
        VmathQuat *result,
        const VmathVector3 *xyz,
        float w
);
```

Arguments

result	The constructed result
xyz	3-D vector
w	Scalar value

Return Values

None

Description

Construct a quaternion with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathQMakeFromV4

Copy elements from a 4-D vector into a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeFromV4(
        VmathQuat *result,
        const VmathVector4 *vec
);
```

Arguments

result	The constructed quaternion
vec	4-D vector

Return Values

None

Description

Construct a quaternion containing the x, y, z, and w elements of the specified 4-D vector.

vmathQMakeIdentity

Construct an identity quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeIdentity(
        VmathQuat *result
);
```

Arguments

result The constructed quaternion

Return Values

None

Description

Construct an identity quaternion equal to (0,0,0,1).

vmathQMakeRotationArc

Construct a quaternion to rotate between two unit-length 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeRotationArc(
        VmathQuat *result,
        const VmathVector3 *unitVec0,
        const VmathVector3 *unitVec1
);
```

Arguments

result	The constructed quaternion
unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a quaternion to rotate between two unit-length 3-D vectors.

Notes

The result is unpredictable if *unitVec0* and *unitVec1* point in opposite directions.

vmathQMakeRotationAxis

Construct a quaternion to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeRotationAxis(
        VmathQuat *result,
        float radians,
        const VmathVector3 *unitVec
);
```

Arguments

result	The constructed quaternion
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a quaternion to rotate around a unit-length 3-D vector by the specified radians angle.

vmathQMakeRotationX

Construct a quaternion to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeRotationX(
        VmathQuat *result,
        float radians
);
```

Arguments

result	The constructed quaternion
radians	Scalar value

Return Values

None

Description

Construct a quaternion to rotate around the x axis by the specified radians angle.

vmathQMakeRotationY

Construct a quaternion to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeRotationY(
        VmathQuat *result,
        float radians
);
```

Arguments

result	The constructed quaternion
radians	Scalar value

Return Values

None

Description

Construct a quaternion to rotate around the y axis by the specified radians angle.

vmathQMakeRotationZ

Construct a quaternion to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMakeRotationZ(
        VmathQuat *result,
        float radians
);
```

Arguments

result	The constructed quaternion
radians	Scalar value

Return Values

None

Description

Construct a quaternion to rotate around the z axis by the specified radians angle.

vmathQMul

Multiply two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQMul(
        VmathQuat *result,
        const VmathQuat *quat0,
        const VmathQuat *quat1
);
```

Arguments

result	Product of the specified quaternions
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Multiply two quaternions.

vmathQNeg

Negate all elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQNeg(
        VmathQuat *result,
        const VmathQuat *quat
);
```

Arguments

result	Quaternion containing negated elements of the specified quaternion
quat	Quaternion

Return Values

None

Description

Negate all elements of a quaternion.

vmathQNorm

Compute the norm of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathQNorm(
        const VmathQuat *quat
);
```

Arguments

quat	Quaternion

Return Values

The norm of the specified quaternion

Description

Compute the norm, equal to the square of the length, of a quaternion.

vmathQNormalize

Normalize a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQNormalize(
        VmathQuat *result,
        const VmathQuat *quat
);
```

Arguments

result	The specified quaternion scaled to unit length
quat	Quaternion

Return Values

None

Description

Compute a normalized quaternion.

Notes

The result is unpredictable when all elements of quat are at or near zero.

vmathQPrint

Print a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQPrint(
        const VmathQuat *quat
);
```

Arguments

quat	Quaternion

Return Values

None

Description

Print a quaternion.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathQPrints

Print a quaternion and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQPrints(
        const VmathQuat *quat,
        const char *name
);
```

Arguments

quat	Quaternion
name	String printed with the quaternion

Return Values

None

Description

Print a quaternion and an associated string identifier.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathQRotate

Use a unit-length quaternion to rotate a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQRotate(
        VmathVector3 *result,
        const VmathQuat *unitQuat,
        const VmathVector3 *vec
);
```

Arguments

result	The rotated 3-D vector, equivalent to unitQuat*Quat(vec,0)*conj(unitQuat)
unitQuat	Quaternion, expected to be unit-length
vec	3-D vector

Return Values

None

Description

Rotate a 3-D vector by applying a unit-length quaternion.

vmathQScalarDiv

Divide a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQScalarDiv(
        VmathQuat *result,
        const VmathQuat *quat,
        float scalar
);
```

Arguments

result	Quotient of the specified quaternion and scalar
quat	Quaternion
scalar	Scalar value

Return Values

None

Description

Divide a quaternion by a scalar.

vmathQScalarMul

Multiply a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQScalarMul(
        VmathQuat *result,
        const VmathQuat *quat,
        float scalar
);
```

Arguments

result	Product of the specified quaternion and scalar
quat	Quaternion
scalar	Scalar value

Return Values

None

Description

Multiply a quaternion by a scalar.

vmathQSelect

Conditionally select between two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSelect(
        VmathQuat *result,
        const VmathQuat *quat0,
        const VmathQuat *quat1,
        unsigned int select1
);
```

Arguments

result	Equal to *quat0* if *select1* == 0, or to *quat1* if *select1* != 0
quat0	Quaternion
quat1	Quaternion
select1	False selects the quat0 argument, true selects the quat1 argument

Return Values

None

Description

Conditionally select one of the quaternion arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathQSetElem

Set an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSetElem(
        VmathQuat *result,
        int idx,
        float value
);
```

Arguments

result	An output quaternion
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathQSetW

Set the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSetW(
        VmathQuat *result,
        float w
);
```

Arguments

result	An output quaternion
w	Scalar value

Return Values

None

Description

Set the w element of a quaternion to the specified scalar value.

vmathQSetX

Set the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSetX(
        VmathQuat *result,
        float x
);
```

Arguments

result	An output quaternion
x	Scalar value

Return Values

None

Description

Set the x element of a quaternion to the specified scalar value.

vmathQSetXYZ

Set the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSetXYZ(
        VmathQuat *result,
        const VmathVector3 *vec
);
```

Arguments

result	An output quaternion
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathQSetY

Set the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSetY(
        VmathQuat *result,
        float y
);
```

Arguments

result	An output quaternion
y	Scalar value

Return Values

None

Description

Set the y element of a quaternion to the specified scalar value.

vmathQSetZ

Set the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSetZ(
        VmathQuat *result,
        float z
);
```

Arguments

result	An output quaternion
z	Scalar value

Return Values

None

Description

Set the z element of a quaternion to the specified scalar value.

vmathQSlerp

Spherical linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSlerp(
        VmathQuat *result,
        float t,
        const VmathQuat *unitQuat0,
        const VmathQuat *unitQuat1
);
```

Arguments

result	Interpolated quaternion
t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length

Return Values

None

Description

Perform spherical linear interpolation between two quaternions.

Notes

Interpolates along the shortest path between orientations. Does not clamp *t* between 0 and 1.

vmathQSquad

Spherical quadrangle interpolation.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSquad(
        VmathQuat *result,
        float t,
        const VmathQuat *unitQuat0,
        const VmathQuat *unitQuat1,
        const VmathQuat *unitQuat2,
        const VmathQuat *unitQuat3
);
```

Arguments

result	Interpolated quaternion
t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length
unitQuat2	Quaternion, expected to be unit-length
unitQuat3	Quaternion, expected to be unit-length

Return Values

None

Description

Perform spherical quadrangle interpolation between four quaternions.

vmathQSub

Subtract a quaternion from another quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathQSub(
        VmathQuat *result,
        const VmathQuat *quat0,
        const VmathQuat *quat1
);
```

Arguments

result	Difference of the specified quaternions
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Subtract a quaternion from another quaternion.

3x3 Matrix Functions (AoS, by reference)

vmathM3AbsPerElem

Compute the absolute value of a 3x3 matrix per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3AbsPerElem(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	3x3 matrix in which each element is the absolute value of the corresponding element of the specified 3x3 matrix
mat	3x3 matrix

Return Values

None

Description

Compute the absolute value of each element of a 3x3 matrix.

vmathM3Add

Add two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Add(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat0,
        const VmathMatrix3 *mat1
);
```

Arguments

result	Sum of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Add two 3x3 matrices.

vmathM3AppendScale

Append (post-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3AppendScale(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat,
        const VmathVector3 *scaleVec
);
```

Arguments

result	The product of `mat` and a scale transformation created from `scaleVec`
mat	3x3 matrix
scaleVec	3-D vector

Return Values

None

Description

Post-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM3Copy

Copy a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Copy(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	The constructed result
mat	3x3 matrix

Return Values

None

Description

Construct a copy of a 3x3 matrix.

vmathM3Determinant

Determinant of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathM3Determinant(
        const VmathMatrix3 *mat
);
```

Arguments

mat 3x3 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 3x3 matrix.

vmathM3GetCol

Get the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3GetCol(
        VmathVector3 *result,
        const VmathMatrix3 *mat,
        int col
);
```

Arguments

result	The column referred to by the specified index
mat	3x3 matrix
col	Index, expected in the range 0-2

Return Values

None

Description

Get the column of a 3x3 matrix referred to by the specified index.

vmathM3GetCol0

Get column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3GetCol0(
        VmathVector3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	Column 0
mat	3x3 matrix

Return Values

None

Description

Get column 0 of a 3x3 matrix.

vmathM3GetCol1

Get column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3GetCol1(
        VmathVector3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	Column 1
mat	3x3 matrix

Return Values

None

Description

Get column 1 of a 3x3 matrix.

vmathM3GetCol2

Get column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3GetCol2(
        VmathVector3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	Column 2
mat	3x3 matrix

Return Values

None

Description

Get column 2 of a 3x3 matrix.

vmathM3GetElem

Get the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathM3GetElem(
        const VmathMatrix3 *mat,
        int col,
        int row
);
```

Arguments

mat	3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x3 matrix referred to by column and row indices.

vmathM3GetRow

Get the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3GetRow(
        VmathVector3 *result,
        const VmathMatrix3 *mat,
        int row
);
```

Arguments

result	The row referred to by the specified index
mat	3x3 matrix
row	Index, expected in the range 0-2

Return Values

None

Description

Get the row of a 3x3 matrix referred to by the specified index.

vmathM3Inverse

Compute the inverse of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Inverse(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	Inverse of *mat*
mat	3x3 matrix

Return Values

None

Description

Compute the inverse of a 3x3 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathM3MakeFromCols

Construct a 3x3 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeFromCols(
        VmathMatrix3 *result,
        const VmathVector3 *col0,
        const VmathVector3 *col1,
        const VmathVector3 *col2
);
```

Arguments

result	The 3x3 matrix that contains the specified columns
col0	3-D vector
col1	3-D vector
col2	3-D vector

Return Values

None

Description

Construct a 3x3 matrix containing the specified columns.

vmathM3MakeFromQ

Construct a 3x3 rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeFromQ(
        VmathMatrix3 *result,
        const VmathQuat *unitQuat
);
```

Arguments

result	A 3x3 matrix that applies the same rotation as *unitQuat*
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathM3MakeFromScalar

Set all elements of a 3x3 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeFromScalar(
        VmathMatrix3 *result,
        float scalar
);
```

Arguments

result	The constructed 3x3 matrix
scalar	Scalar value

Return Values

None

Description

Construct a 3x3 matrix with all elements set to the scalar value argument.

vmathM3MakeIdentity

Construct an identity 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeIdentity(
        VmathMatrix3 *result
);
```

Arguments

`result` The constructed 3x3 matrix

Return Values

None

Description

Construct an identity 3x3 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathM3MakeRotationAxis

Construct a 3x3 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeRotationAxis(
        VmathMatrix3 *result,
        float radians,
        const VmathVector3 *unitVec
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a 3x3 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathM3MakeRotationQ

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeRotationQ(
        VmathMatrix3 *result,
        const VmathQuat *unitQuat
);
```

Arguments

result	A 3x3 matrix that applies the same rotation as *unitQuat*
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathM3MakeRotationX

Construct a 3x3 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeRotationX(
        VmathMatrix3 *result,
        float radians
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x3 matrix to rotate around the x axis by the specified radians angle.

vmathM3MakeRotationY

Construct a 3x3 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeRotationY(
        VmathMatrix3 *result,
        float radians
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x3 matrix to rotate around the y axis by the specified radians angle.

vmathM3MakeRotationZ

Construct a 3x3 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeRotationZ(
        VmathMatrix3 *result,
        float radians
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x3 matrix to rotate around the z axis by the specified radians angle.

vmathM3MakeRotationZYX

Construct a 3x3 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeRotationZYX(
        VmathMatrix3 *result,
        const VmathVector3 *radiansXYZ
);
```

Arguments

result	The constructed 3x3 matrix
radiansXYZ	3-D vector

Return Values

None

Description

Construct a 3x3 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to `rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())`.

vmathM3MakeScale

Construct a 3x3 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MakeScale(
        VmathMatrix3 *result,
        const VmathVector3 *scaleVec
);
```

Arguments

result	The constructed 3x3 matrix
scaleVec	3-D vector

Return Values

None

Description

Construct a 3x3 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathM3Mul

Multiply two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Mul(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat0,
        const VmathMatrix3 *mat1
);
```

Arguments

result	Product of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Multiply two 3x3 matrices.

vmathM3MulPerElem

Multiply two 3x3 matrices per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MulPerElem(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat0,
        const VmathMatrix3 *mat1
);
```

Arguments

result	3x3 matrix in which each element is the product of the corresponding elements of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Multiply two 3x3 matrices element by element.

vmathM3MulV3

Multiply a 3x3 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3MulV3(
        VmathVector3 *result,
        const VmathMatrix3 *mat,
        const VmathVector3 *vec
);
```

Arguments

result	Product of the specified 3x3 matrix and 3-D vector
mat	3x3 matrix
vec	3-D vector

Return Values

None

Description

Multiply a 3x3 matrix by a 3-D vector.

vmathM3Neg

Negate all elements of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Neg(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	3x3 matrix containing negated elements of the specified 3x3 matrix
mat	3x3 matrix

Return Values

None

Description

Negate all elements of a 3x3 matrix.

vmathM3PrependScale

Prepend (pre-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3PrependScale(
        VmathMatrix3 *result,
        const VmathVector3 *scaleVec,
        const VmathMatrix3 *mat
);
```

Arguments

result	The product of a scale transformation created from *scaleVec* and *mat*
scaleVec	3-D vector
mat	3x3 matrix

Return Values

None

Description

Pre-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM3Print

Print a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Print(
        const VmathMatrix3 *mat
);
```

Arguments

mat 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM3Prints

Print a 3x3 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Prints(
        const VmathMatrix3 *mat,
        const char *name
);
```

Arguments

mat	3x3 matrix
name	String printed with the 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix and an associated string identifier. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM3ScalarMul

Multiply a 3x3 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3ScalarMul(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat,
        float scalar
);
```

Arguments

result	Product of the specified 3x3 matrix and scalar
mat	3x3 matrix
scalar	Scalar value

Return Values

None

Description

Multiply a 3x3 matrix by a scalar.

vmathM3Select

Conditionally select between two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Select(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat0,
        const VmathMatrix3 *mat1,
        unsigned int select1
);
```

Arguments

result	Equal to `mat0` if `select1` == 0, or to `mat1` if `select1` != 0
mat0	3x3 matrix
mat1	3x3 matrix
select1	False selects the mat0 argument, true selects the mat1 argument

Return Values

None

Description

Conditionally select one of the 3x3 matrix arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of `select1` to a VMX register may use more processing time than a branch.

vmathM3SetCol

Set the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3SetCol(
        VmathMatrix3 *result,
        int col,
        const VmathVector3 *vec
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the column of a 3x3 matrix referred to by the specified index.

vmathM3SetCol0

Set column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3SetCol0(
        VmathMatrix3 *result,
        const VmathVector3 *col0
);
```

Arguments

result	An output 3x3 matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x3 matrix.

vmathM3SetCol1

Set column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3SetCol1(
        VmathMatrix3 *result,
        const VmathVector3 *col1
);
```

Arguments

result	An output 3x3 matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x3 matrix.

vmathM3SetCol2

Set column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3SetCol2(
        VmathMatrix3 *result,
        const VmathVector3 *col2
);
```

Arguments

result	An output 3x3 matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x3 matrix.

vmathM3SetElem

Set the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3SetElem(
        VmathMatrix3 *result,
        int col,
        int row,
        float val
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x3 matrix referred to by column and row indices.

vmathM3SetRow

Set the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3SetRow(
        VmathMatrix3 *result,
        int row,
        const VmathVector3 *vec
);
```

Arguments

result	An output 3x3 matrix
row	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the row of a 3x3 matrix referred to by the specified index.

vmathM3Sub

Subtract a 3x3 matrix from another 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Sub(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat0,
        const VmathMatrix3 *mat1
);
```

Arguments

result	Difference of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Subtract a 3x3 matrix from another 3x3 matrix.

vmathM3Transpose

Transpose of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM3Transpose(
        VmathMatrix3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	*mat* transposed
mat	3x3 matrix

Return Values

None

Description

Compute the transpose of a 3x3 matrix.

4x4 Matrix Functions (AoS, by reference)

vmathM4AbsPerElem

Compute the absolute value of a 4x4 matrix per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4AbsPerElem(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	4x4 matrix in which each element is the absolute value of the corresponding element of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Compute the absolute value of each element of a 4x4 matrix.

vmathM4Add

Add two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Add(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat0,
        const VmathMatrix4 *mat1
);
```

Arguments

result	Sum of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Add two 4x4 matrices.

vmathM4AffineInverse

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4AffineInverse(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Inverse of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is inverse(M), whose translation vector is -inverse(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions. The result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathM4AppendScale

Append (post-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4AppendScale(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat,
        const VmathVector3 *scaleVec
);
```

Arguments

result	The product of `mat` and a scale transformation created from `scaleVec`
mat	4x4 matrix
scaleVec	3-D vector

Return Values

None

Description

Post-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM4Copy

Copy a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Copy(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	The constructed result
mat	4x4 matrix

Return Values

None

Description

Construct a copy of a 4x4 matrix.

vmathM4Determinant

Determinant of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathM4Determinant(
        const VmathMatrix4 *mat
);
```

Arguments

mat 4x4 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 4x4 matrix.

vmathM4GetCol

Get the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetCol(
        VmathVector4 *result,
        const VmathMatrix4 *mat,
        int col
);
```

Arguments

result	The column referred to by the specified index
mat	4x4 matrix
col	Index, expected in the range 0-3

Return Values

None

Description

Get the column of a 4x4 matrix referred to by the specified index.

vmathM4GetCol0

Get column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetCol0(
        VmathVector4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Column 0
mat	4x4 matrix

Return Values

None

Description

Get column 0 of a 4x4 matrix.

vmathM4GetCol1

Get column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetCol1(
        VmathVector4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Column 1
mat	4x4 matrix

Return Values

None

Description

Get column 1 of a 4x4 matrix.

vmathM4GetCol2

Get column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetCol2(
        VmathVector4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Column 2
mat	4x4 matrix

Return Values

None

Description

Get column 2 of a 4x4 matrix.

vmathM4GetCol3

Get column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetCol3(
        VmathVector4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Column 3
mat	4x4 matrix

Return Values

None

Description

Get column 3 of a 4x4 matrix.

vmathM4GetElem

Get the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathM4GetElem(
        const VmathMatrix4 *mat,
        int col,
        int row
);
```

Arguments

mat	4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3

Return Values

Element selected by *col* and *row*

Description

Get the element of a 4x4 matrix referred to by column and row indices.

vmathM4GetRow

Get the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetRow(
        VmathVector4 *result,
        const VmathMatrix4 *mat,
        int row
);
```

Arguments

result	The row referred to by the specified index
mat	4x4 matrix
row	Index, expected in the range 0-3

Return Values

None

Description

Get the row of a 4x4 matrix referred to by the specified index.

vmathM4GetTranslation

Get the translation component of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetTranslation(
        VmathVector3 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Translation component
mat	4x4 matrix

Return Values

None

Description

Get the translation component of a 4x4 matrix.

vmathM4GetUpper3x3

Get the upper-left 3x3 submatrix of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4GetUpper3x3(
        VmathMatrix3 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Upper-left 3x3 submatrix
mat	4x4 matrix

Return Values

None

Description

Get the upper-left 3x3 submatrix of a 4x4 matrix.

vmathM4Inverse

Compute the inverse of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Inverse(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Inverse of *mat*
mat	4x4 matrix

Return Values

None

Description

Compute the inverse of a 4x4 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathM4MakeFromCols

Construct a 4x4 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeFromCols(
        VmathMatrix4 *result,
        const VmathVector4 *col0,
        const VmathVector4 *col1,
        const VmathVector4 *col2,
        const VmathVector4 *col3
);
```

Arguments

result	The 4x4 matrix that contains the specified columns
col0	4-D vector
col1	4-D vector
col2	4-D vector
col3	4-D vector

Return Values

None

Description

Construct a 4x4 matrix containing the specified columns.

vmathM4MakeFromM3V3

Construct a 4x4 matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeFromM3V3(
        VmathMatrix4 *result,
        const VmathMatrix3 *mat,
        const VmathVector3 *translateVec
);
```

Arguments

result	The constructed 4x4 matrix
mat	3x3 matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix whose upper 3x3 elements are equal to the 3x3 matrix argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathM4MakeFromQV3

Construct a 4x4 matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeFromQV3(
        VmathMatrix4 *result,
        const VmathQuat *unitQuat,
        const VmathVector3 *translateVec
);
```

Arguments

result	The constructed 4x4 matrix
unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathM4MakeFromScalar

Set all elements of a 4x4 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeFromScalar(
        VmathMatrix4 *result,
        float scalar
);
```

Arguments

result	The constructed 4x4 matrix
scalar	Scalar value

Return Values

None

Description

Construct a 4x4 matrix with all elements set to the scalar value argument.

vmathM4MakeFromT3

Construct a 4x4 matrix from a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeFromT3(
        VmathMatrix4 *result,
        const VmathTransform3 *mat
);
```

Arguments

result	The constructed 4x4 matrix
mat	3x4 transformation matrix

Return Values

None

Description

Construct a 4x4 matrix whose upper 3x4 elements are equal to the 3x4 transformation matrix argument and whose bottom row is equal to (0,0,0,1).

vmathM4MakeFrustum

Construct a perspective projection matrix based on frustum.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeFrustum(
        VmathMatrix4 *result,
        float left,
        float right,
        float bottom,
        float top,
        float zNear,
        float zFar
);
```

Arguments

result	The constructed 4x4 matrix
left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

None

Description

Construct a perspective projection matrix based on frustum, equal to:

```
2*zNear/(right-left)   0          (right+left)/(right-left)      0
       0      2*zNear/(top-bottom)  (top+bottom)/(top-bottom)      0
       0               0        -(zFar+zNear)/(zFar-zNear)
-2*zFar*zNear/(zFar-zNear)
       0               0                  -1                0  .
```

vmathM4MakeIdentity

Construct an identity 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeIdentity(
        VmathMatrix4 *result
);
```

Arguments

result The constructed 4x4 matrix

Return Values

None

Description

Construct an identity 4x4 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathM4MakeLookAt

Construct viewing matrix based on eye position, position looked at, and up direction.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeLookAt(
        VmathMatrix4 *result,
        const VmathPoint3 *eyePos,
        const VmathPoint3 *lookAtPos,
        const VmathVector3 *upVec
);
```

Arguments

result	The constructed 4x4 matrix
eyePos	3-D point
lookAtPos	3-D point
upVec	3-D vector

Return Values

None

Description

Construct the inverse of a coordinate frame that is centered at the eye position, with z axis directed away from lookAtPos, and y axis oriented to best match the up direction.

vmathM4MakeOrthographic

Construct an orthographic projection matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeOrthographic(
        VmathMatrix4 *result,
        float left,
        float right,
        float bottom,
        float top,
        float zNear,
        float zFar
);
```

Arguments

result	The constructed 4x4 matrix
left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

None

Description

Construct an orthographic projection matrix, equal to

```
2/(right-left)        0              0          -(right+left)/(right-left)
      0         2/(top-bottom)       0          -(top+bottom)/(top-bottom)
      0               0        -2/(zFar-zNear)  -(zFar+zNear)/(zFar-zNear)
      0               0              0                    1  .
```

vmathM4MakePerspective

Construct a perspective projection matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakePerspective(
        VmathMatrix4 *result,
        float fovyRadians,
        float aspect,
        float zNear,
        float zFar
);
```

Arguments

result	The constructed 4x4 matrix
fovyRadians	Scalar value
aspect	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

None

Description

Construct a perspective projection matrix, equal to:

```
cot(fovyRadians/2)/aspect   0                  0                        0
       0             cot(fovyRadians/2)       0                        0
       0                       0   (zFar+zNear)/(zNear-zFar)
2*zFar*zNear/(zNear-zFar)
       0                       0              -1                       0 .
```

vmathM4MakeRotationAxis

Construct a 4x4 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeRotationAxis(
        VmathMatrix4 *result,
        float radians,
        const VmathVector3 *unitVec
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a 4x4 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathM4MakeRotationQ

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeRotationQ(
        VmathMatrix4 *result,
        const VmathQuat *unitQuat
);
```

Arguments

result	A 4x4 matrix that applies the same rotation as *unitQuat*
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 4x4 matrix that applies the same rotation as the specified unit-length quaternion.

vmathM4MakeRotationX

Construct a 4x4 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeRotationX(
        VmathMatrix4 *result,
        float radians
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value

Return Values

None

Description

Construct a 4x4 matrix to rotate around the x axis by the specified radians angle.

vmathM4MakeRotationY

Construct a 4x4 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeRotationY(
        VmathMatrix4 *result,
        float radians
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value

Return Values

None

Description

Construct a 4x4 matrix to rotate around the y axis by the specified radians angle.

vmathM4MakeRotationZ

Construct a 4x4 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeRotationZ(
        VmathMatrix4 *result,
        float radians
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value

Return Values

None

Description

Construct a 4x4 matrix to rotate around the z axis by the specified radians angle.

vmathM4MakeRotationZYX

Construct a 4x4 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeRotationZYX(
        VmathMatrix4 *result,
        const VmathVector3 *radiansXYZ
);
```

Arguments

result	The constructed 4x4 matrix
radiansXYZ	3-D vector

Return Values

None

Description

Construct a 4x4 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to `rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())`.

vmathM4MakeScale

Construct a 4x4 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeScale(
        VmathMatrix4 *result,
        const VmathVector3 *scaleVec
);
```

Arguments

result	The constructed 4x4 matrix
scaleVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathM4MakeTranslation

Construct a 4x4 matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MakeTranslation(
        VmathMatrix4 *result,
        const VmathVector3 *translateVec
);
```

Arguments

result	The constructed 4x4 matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathM4Mul

Multiply two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Mul(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat0,
        const VmathMatrix4 *mat1
);
```

Arguments

result	Product of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Multiply two 4x4 matrices.

vmathM4MulP3

Multiply a 4x4 matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MulP3(
        VmathVector4 *result,
        const VmathMatrix4 *mat,
        const VmathPoint3 *pnt
);
```

Arguments

result	Product of the specified 4x4 matrix and 3-D point
mat	4x4 matrix
pnt	3-D point

Return Values

None

Description

Multiply a 4x4 matrix by a 3-D point treated as if it were a 4-D vector with the w element equal to 1.

vmathM4MulPerElem

Multiply two 4x4 matrices per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MulPerElem(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat0,
        const VmathMatrix4 *mat1
);
```

Arguments

result	4x4 matrix in which each element is the product of the corresponding elements of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Multiply two 4x4 matrices element by element.

vmathM4MulT3

Multiply a 4x4 matrix by a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MulT3(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Product of the specified 4x4 matrix and 3x4 transformation matrix
mat	4x4 matrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Multiply a 4x4 matrix by a 3x4 transformation matrix treated as if it were a 4x4 matrix with the bottom row equal to (0,0,0,1).

vmathM4MulV3

Multiply a 4x4 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MulV3(
        VmathVector4 *result,
        const VmathMatrix4 *mat,
        const VmathVector3 *vec
);
```

Arguments

result	Product of the specified 4x4 matrix and 3-D vector
mat	4x4 matrix
vec	3-D vector

Return Values

None

Description

Multiply a 4x4 matrix by a 3-D vector treated as if it were a 4-D vector with the w element equal to 0.

vmathM4MulV4

Multiply a 4x4 matrix by a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4MulV4(
        VmathVector4 *result,
        const VmathMatrix4 *mat,
        const VmathVector4 *vec
);
```

Arguments

result	Product of the specified 4x4 matrix and 4-D vector
mat	4x4 matrix
vec	4-D vector

Return Values

None

Description

Multiply a 4x4 matrix by a 4-D vector.

vmathM4Neg

Negate all elements of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Neg(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	4x4 matrix containing negated elements of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Negate all elements of a 4x4 matrix.

vmathM4OrthoInverse

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix with an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4OrthoInverse(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	Inverse of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), whose translation vector is -transpose(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions.

vmathM4PrependScale

Prepend (pre-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4PrependScale(
        VmathMatrix4 *result,
        const VmathVector3 *scaleVec,
        const VmathMatrix4 *mat
);
```

Arguments

result	The product of a scale transformation created from *scaleVec* and *mat*
scaleVec	3-D vector
mat	4x4 matrix

Return Values

None

Description

Pre-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM4Print

Print a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Print(
        const VmathMatrix4 *mat
);
```

Arguments

mat 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM4Prints

Print a 4x4 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Prints(
        const VmathMatrix4 *mat,
        const char *name
);
```

Arguments

mat	4x4 matrix
name	String printed with the 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix and an associated string identifier. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM4ScalarMul

Multiply a 4x4 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4ScalarMul(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat,
        float scalar
);
```

Arguments

result	Product of the specified 4x4 matrix and scalar
mat	4x4 matrix
scalar	Scalar value

Return Values

None

Description

Multiply a 4x4 matrix by a scalar.

vmathM4Select

Conditionally select between two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Select(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat0,
        const VmathMatrix4 *mat1,
        unsigned int select1
);
```

Arguments

result	Equal to *mat0* if *select1* == 0, or to *mat1* if *select1* != 0
mat0	4x4 matrix
mat1	4x4 matrix
select1	False selects the mat0 argument, true selects the mat1 argument

Return Values

None

Description

Conditionally select one of the 4x4 matrix arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathM4SetCol

Set the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetCol(
        VmathMatrix4 *result,
        int col,
        const VmathVector4 *vec
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the column of a 4x4 matrix referred to by the specified index.

vmathM4SetCol0

Set column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetCol0(
        VmathMatrix4 *result,
        const VmathVector4 *col0
);
```

Arguments

result	An output 4x4 matrix
col0	4-D vector

Return Values

None

Description

Set column 0 of a 4x4 matrix.

vmathM4SetCol1

Set column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetCol1(
        VmathMatrix4 *result,
        const VmathVector4 *col1
);
```

Arguments

result	An output 4x4 matrix
col1	4-D vector

Return Values

None

Description

Set column 1 of a 4x4 matrix.

vmathM4SetCol2

Set column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetCol2(
        VmathMatrix4 *result,
        const VmathVector4 *col2
);
```

Arguments

result	An output 4x4 matrix
col2	4-D vector

Return Values

None

Description

Set column 2 of a 4x4 matrix.

vmathM4SetCol3

Set column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetCol3(
        VmathMatrix4 *result,
        const VmathVector4 *col3
);
```

Arguments

result	An output 4x4 matrix
col3	4-D vector

Return Values

None

Description

Set column 3 of a 4x4 matrix.

vmathM4SetElem

Set the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetElem(
        VmathMatrix4 *result,
        int col,
        int row,
        float val
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3
val	Scalar value

Return Values

None

Description

Set the element of a 4x4 matrix referred to by column and row indices.

vmathM4SetRow

Set the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetRow(
        VmathMatrix4 *result,
        int row,
        const VmathVector4 *vec
);
```

Arguments

result	An output 4x4 matrix
row	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the row of a 4x4 matrix referred to by the specified index.

vmathM4SetTranslation

Set translation component.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetTranslation(
        VmathMatrix4 *result,
        const VmathVector3 *translateVec
);
```

Arguments

result	An output 4x4 matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 4x4 matrix equal to the specified 3-D vector.

Notes

This function does not change the bottom row elements.

vmathM4SetUpper3x3

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4SetUpper3x3(
        VmathMatrix4 *result,
        const VmathMatrix3 *mat3
);
```

Arguments

result	An output 4x4 matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 4x4 matrix equal to the specified 3x3 matrix.

Notes

This function does not change the bottom row elements.

vmathM4Sub

Subtract a 4x4 matrix from another 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Sub(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat0,
        const VmathMatrix4 *mat1
);
```

Arguments

result	Difference of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Subtract a 4x4 matrix from another 4x4 matrix.

vmathM4Transpose

Transpose of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathM4Transpose(
        VmathMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	*mat* transposed
mat	4x4 matrix

Return Values

None

Description

Compute the transpose of a 4x4 matrix.

Transformation Functions (AoS, by reference)

vmathT3AbsPerElem

Compute the absolute value of a 3x4 transformation matrix per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3AbsPerElem(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	3x4 transformation matrix in which each element is the absolute value of the corresponding element of the specified 3x4 transformation matrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Compute the absolute value of each element of a 3x4 transformation matrix.

vmathT3AppendScale

Append (post-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3AppendScale(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm,
        const VmathVector3 *scaleVec
);
```

Arguments

result	The product of *tfrm* and a scale transformation created from *scaleVec*
tfrm	3x4 transformation matrix
scaleVec	3-D vector

Return Values

None

Description

Post-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathT3Copy

Copy a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3Copy(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	The constructed result
tfrm	3x4 transformation matrix

Return Values

None

Description

Construct a copy of a 3x4 transformation matrix.

vmathT3GetCol

Get the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetCol(
        VmathVector3 *result,
        const VmathTransform3 *tfrm,
        int col
);
```

Arguments

result	The column referred to by the specified index
tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3

Return Values

None

Description

Get the column of a 3x4 transformation matrix referred to by the specified index.

vmathT3GetCol0

Get column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetCol0(
        VmathVector3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Column 0
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 0 of a 3x4 transformation matrix.

vmathT3GetCol1

Get column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetCol1(
        VmathVector3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Column 1
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 1 of a 3x4 transformation matrix.

vmathT3GetCol2

Get column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetCol2(
        VmathVector3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Column 2
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 2 of a 3x4 transformation matrix.

vmathT3GetCol3

Get column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetCol3(
        VmathVector3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Column 3
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 3 of a 3x4 transformation matrix.

vmathT3GetElem

Get the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline float vmathT3GetElem(
        const VmathTransform3 *tfrm,
        int col,
        int row
);
```

Arguments

tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x4 transformation matrix referred to by column and row indices.

vmathT3GetRow

Get the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetRow(
        VmathVector4 *result,
        const VmathTransform3 *tfrm,
        int row
);
```

Arguments

result	The row referred to by the specified index
tfrm	3x4 transformation matrix
row	Index, expected in the range 0-2

Return Values

None

Description

Get the row of a 3x4 transformation matrix referred to by the specified index.

vmathT3GetTranslation

Get the translation component of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetTranslation(
        VmathVector3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Translation component
tfrm	3x4 transformation matrix

Return Values

None

Description

Get the translation component of a 3x4 transformation matrix.

vmathT3GetUpper3x3

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3GetUpper3x3(
        VmathMatrix3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Upper-left 3x3 submatrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

vmathT3Inverse

Inverse of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3Inverse(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Inverse of *tfrm*
tfrm	3x4 transformation matrix

Return Values

None

Description

Compute the inverse of a 3x4 transformation matrix.

Notes

Result is unpredictable when the determinant of the left 3x3 submatrix is equal to or near 0.

vmathT3MakeFromCols

Construct a 3x4 transformation matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeFromCols(
        VmathTransform3 *result,
        const VmathVector3 *col0,
        const VmathVector3 *col1,
        const VmathVector3 *col2,
        const VmathVector3 *col3
);
```

Arguments

result	The 3x4 transformation matrix that contains the specified columns
col0	3-D vector
col1	3-D vector
col2	3-D vector
col3	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix containing the specified columns.

vmathT3MakeFromM3V3

Construct a 3x4 transformation matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeFromM3V3(
        VmathTransform3 *result,
        const VmathMatrix3 *tfrm,
        const VmathVector3 *translateVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
tfrm	3x3 matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix whose upper 3x3 elements are equal to the 3x3 matrix argument and whose translation component is equal to the 3-D vector argument.

vmathT3MakeFromQV3

Construct a 3x4 transformation matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeFromQV3(
        VmathTransform3 *result,
        const VmathQuat *unitQuat,
        const VmathVector3 *translateVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument and whose translation component is equal to the 3-D vector argument.

vmathT3MakeFromScalar

Set all elements of a 3x4 transformation matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeFromScalar(
        VmathTransform3 *result,
        float scalar
);
```

Arguments

result	The constructed 3x4 transformation matrix
scalar	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix with all elements set to the scalar value argument.

vmathT3MakeIdentity

Construct an identity 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeIdentity(
        VmathTransform3 *result
);
```

Arguments

result The constructed 3x4 transformation matrix

Return Values

None

Description

Construct an identity 3x4 transformation matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathT3MakeRotationAxis

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeRotationAxis(
        VmathTransform3 *result,
        float radians,
        const VmathVector3 *unitVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathT3MakeRotationQ

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeRotationQ(
        VmathTransform3 *result,
        const VmathQuat *unitQuat
);
```

Arguments

result	A 3x4 transformation matrix that applies the same rotation as *unitQuat*
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 3x4 transformation matrix that applies the same rotation as the specified unit-length quaternion.

vmathT3MakeRotationX

Construct a 3x4 transformation matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeRotationX(
        VmathTransform3 *result,
        float radians
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the x axis by the specified radians angle.

vmathT3MakeRotationY

Construct a 3x4 transformation matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeRotationY(
        VmathTransform3 *result,
        float radians
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the y axis by the specified radians angle.

vmathT3MakeRotationZ

Construct a 3x4 transformation matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeRotationZ(
        VmathTransform3 *result,
        float radians
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the z axis by the specified radians angle.

vmathT3MakeRotationZYX

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeRotationZYX(
        VmathTransform3 *result,
        const VmathVector3 *radiansXYZ
);
```

Arguments

result	The constructed 3x4 transformation matrix
radiansXYZ	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to `rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())`.

vmathT3MakeScale

Construct a 3x4 transformation matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeScale(
        VmathTransform3 *result,
        const VmathVector3 *scaleVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
scaleVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathT3MakeTranslation

Construct a 3x4 transformation matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MakeTranslation(
        VmathTransform3 *result,
        const VmathVector3 *translateVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in `translateVec`.

vmathT3Mul

Multiply two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3Mul(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm0,
        const VmathTransform3 *tfrm1
);
```

Arguments

result	Product of the specified 3x4 transformation matrices
tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix

Return Values

None

Description

Multiply two 3x4 transformation matrices.

vmathT3MulP3

Multiply a 3x4 transformation matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MulP3(
        VmathPoint3 *result,
        const VmathTransform3 *tfrm,
        const VmathPoint3 *pnt
);
```

Arguments

result	Product of the specified 3x4 transformation matrix and 3-D point
tfrm	3x4 transformation matrix
pnt	3-D point

Return Values

None

Description

Applies the 3x3 upper-left submatrix and the translation component of a 3x4 transformation matrix to a 3-D point.

vmathT3MulPerElem

Multiply two 3x4 transformation matrices per element.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MulPerElem(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm0,
        const VmathTransform3 *tfrm1
);
```

Arguments

result	3x4 transformation matrix in which each element is the product of the corresponding elements of the specified 3x4 transformation matrices
tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix

Return Values

None

Description

Multiply two 3x4 transformation matrices element by element.

vmathT3MulV3

Multiply a 3x4 transformation matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3MulV3(
        VmathVector3 *result,
        const VmathTransform3 *tfrm,
        const VmathVector3 *vec
);
```

Arguments

result	Product of the specified 3x4 transformation matrix and 3-D vector
tfrm	3x4 transformation matrix
vec	3-D vector

Return Values

None

Description

Applies the 3x3 upper-left submatrix (but not the translation component) of a 3x4 transformation matrix to a 3-D vector.

vmathT3OrthoInverse

Compute the inverse of a 3x4 transformation matrix, expected to have an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3OrthoInverse(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	Inverse of the specified 3x4 transformation matrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Naming the upper-left 3x3 submatrix of the specified 3x4 transformation matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), and whose translation vector is -transpose(M)*v.

Notes

This can be used to achieve better performance than a general inverse when the specified 3x4 transformation matrix meets the given restrictions.

vmathT3PrependScale

Prepend (pre-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3PrependScale(
        VmathTransform3 *result,
        const VmathVector3 *scaleVec,
        const VmathTransform3 *tfrm
);
```

Arguments

result	The product of a scale transformation created from *scaleVec* and *tfrm*
scaleVec	3-D vector
tfrm	3x4 transformation matrix

Return Values

None

Description

Pre-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathT3Print

Print a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3Print(
        const VmathTransform3 *tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathT3Prints

Print a 3x4 transformation matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3Prints(
        const VmathTransform3 *tfrm,
        const char *name
);
```

Arguments

tfrm	3x4 transformation matrix
name	String printed with the 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix and an associated string identifier. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathT3Select

Conditionally select between two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3Select(
        VmathTransform3 *result,
        const VmathTransform3 *tfrm0,
        const VmathTransform3 *tfrm1,
        unsigned int select1
);
```

Arguments

result	Equal to *tfrm0* if *select1* == 0, or to *tfrm1* if *select1* != 0
tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix
select1	False selects the tfrm0 argument, true selects the tfrm1 argument

Return Values

None

Description

Conditionally select one of the 3x4 transformation matrix arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathT3SetCol

Set the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetCol(
        VmathTransform3 *result,
        int col,
        const VmathVector3 *vec
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
vec	3-D vector

Return Values

None

Description

Set the column of a 3x4 transformation matrix referred to by the specified index.

vmathT3SetCol0

Set column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetCol0(
        VmathTransform3 *result,
        const VmathVector3 *col0
);
```

Arguments

result	An output 3x4 transformation matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x4 transformation matrix.

vmathT3SetCol1

Set column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetCol1(
        VmathTransform3 *result,
        const VmathVector3 *col1
);
```

Arguments

result	An output 3x4 transformation matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x4 transformation matrix.

vmathT3SetCol2

Set column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetCol2(
        VmathTransform3 *result,
        const VmathVector3 *col2
);
```

Arguments

result	An output 3x4 transformation matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x4 transformation matrix.

vmathT3SetCol3

Set column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetCol3(
        VmathTransform3 *result,
        const VmathVector3 *col3
);
```

Arguments

result	An output 3x4 transformation matrix
col3	3-D vector

Return Values

None

Description

Set column 3 of a 3x4 transformation matrix.

vmathT3SetElem

Set the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetElem(
        VmathTransform3 *result,
        int col,
        int row,
        float val
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x4 transformation matrix referred to by column and row indices.

vmathT3SetRow

Set the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetRow(
        VmathTransform3 *result,
        int row,
        const VmathVector4 *vec
);
```

Arguments

result	An output 3x4 transformation matrix
row	Index, expected in the range 0-2
vec	4-D vector

Return Values

None

Description

Set the row of a 3x4 transformation matrix referred to by the specified index.

vmathT3SetTranslation

Set translation component.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetTranslation(
        VmathTransform3 *result,
        const VmathVector3 *translateVec
);
```

Arguments

result	An output 3x4 transformation matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 3x4 transformation matrix equal to the specified 3-D vector.

vmathT3SetUpper3x3

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos.h>
static inline void vmathT3SetUpper3x3(
        VmathTransform3 *result,
        const VmathMatrix3 *mat3
);
```

Arguments

result	An output 3x4 transformation matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 3x4 transformation matrix equal to the specified 3x3 matrix.

3-D Vector Functions (SoA, by reference)

vmathSoaV3AbsPerElem

Compute the absolute value of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3AbsPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the absolute value of the corresponding element of vec
vec	3-D vector

Return Values

None

Description

Compute the absolute value of each element of a 3-D vector.

vmathSoaV3Add

Add two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Add(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	Sum of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Add two 3-D vectors.

vmathSoaV3AddP3

Add a 3-D vector to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3AddP3(
        VmathSoaPoint3 *result,
        const VmathSoaVector3 *vec,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	Sum of the specified 3-D vector and 3-D point
vec	3-D vector
pnt	3-D point

Return Values

None

Description

Add a 3-D vector to a 3-D point.

vmathSoaV3Copy

Copy a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Copy(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	The constructed result
vec	3-D vector

Return Values

None

Description

Construct a copy of a 3-D vector.

vmathSoaV3CopySignPerElem

Copy sign from one 3-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3CopySignPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	3-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathSoaV3Cross

Compute cross product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Cross(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	Cross product of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Compute cross product of two 3-D vectors.

vmathSoaV3CrossMatrix

Cross-product matrix of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3CrossMatrix(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	Cross-product matrix of *vec*
vec	3-D vector

Return Values

None

Description

Compute a matrix that, when multiplied by a 3-D vector, produces the same result as a cross product with that 3-D vector.

vmathSoaV3CrossMatrixMul

Create cross-product matrix and multiply.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3CrossMatrixMul(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *vec,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	Product of cross-product matrix of *vec* and *mat*
vec	3-D vector
mat	3x3 matrix

Return Values

None

Description

Multiply a cross-product matrix by another matrix.

Notes

Faster than separately creating a cross-product matrix and multiplying.

vmathSoaV3DivPerElem

Divide two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3DivPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the quotient of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Divide two 3-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathSoaV3Dot

Compute the dot product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3Dot(
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Dot product of the specified 3-D vectors

Description

Compute the dot product of two 3-D vectors.

vmathSoaV3Get4Aos

Extract four AoS 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Get4Aos(
        const VmathSoaVector3 *vec,
        VmathVector3 *result0,
        VmathVector3 *result1,
        VmathVector3 *result2,
        VmathVector3 *result3
);
```

Arguments

vec	3-D vector
result0	An output AoS 3-D vector
result1	An output AoS 3-D vector
result2	An output AoS 3-D vector
result3	An output AoS 3-D vector

Return Values

None

Description

Extract four AoS 3-D vectors from four slots of an SoA 3-D vector (transpose the data format).

vmathSoaV3GetElem

Get an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3GetElem(
        const VmathSoaVector3 *vec,
        int idx
);
```

Arguments

vec	3-D vector
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathSoaV3GetX

Get the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3GetX(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

x element of a 3-D vector

Description

Get the x element of a 3-D vector.

vmathSoaV3GetY

Get the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3GetY(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

y element of a 3-D vector

Description

Get the y element of a 3-D vector.

vmathSoaV3GetZ

Get the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3GetZ(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

z element of a 3-D vector

Description

Get the z element of a 3-D vector.

vmathSoaV3Length

Compute the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3Length(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Length of the specified 3-D vector

Description

Compute the length of a 3-D vector.

vmathSoaV3LengthSqr

Compute the square of the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3LengthSqr(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Square of the length of the specified 3-D vector

Description

Compute the square of the length of a 3-D vector.

vmathSoaV3Lerp

Linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Lerp(
        VmathSoaVector3 *result,
        vec_float4 t,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	Interpolated 3-D vector
t	Interpolation parameter
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Linearly interpolate between two 3-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaV3LoadXYZArray

Load four three-float 3-D vectors, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3LoadXYZArray(
        VmathSoaVector3 *vec,
        const vec_float4 *threeQuads
);
```

Arguments

vec	An output 3-D vector
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D vectors, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four slots of an SoA 3-D vector.

vmathSoaV3MakeFrom4Aos

Insert four AoS 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeFrom4Aos(
        VmathSoaVector3 *result,
        const VmathVector3 *vec0,
        const VmathVector3 *vec1,
        const VmathVector3 *vec2,
        const VmathVector3 *vec3
);
```

Arguments

result	The constructed SoA 3-D vector
vec0	AoS 3-D vector
vec1	AoS 3-D vector
vec2	AoS 3-D vector
vec3	AoS 3-D vector

Return Values

None

Description

Insert four AoS 3-D vectors into four slots of an SoA 3-D vector (transpose the data format).

vmathSoaV3MakeFromAos

Replicate an AoS 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeFromAos(
        VmathSoaVector3 *result,
        const VmathVector3 *vec
);
```

Arguments

result	The constructed SoA 3-D vector
vec	AoS 3-D vector

Return Values

None

Description

Replicate an AoS 3-D vector in all four slots of an SoA 3-D vector.

vmathSoaV3MakeFromElems

Construct a 3-D vector from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeFromElems(
        VmathSoaVector3 *result,
        vec_float4 x,
        vec_float4 y,
        vec_float4 z
);
```

Arguments

result	The 3-D vector that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value

Return Values

None

Description

Construct a 3-D vector containing the specified x, y, and z elements.

vmathSoaV3MakeFromP3

Copy elements from a 3-D point into a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeFromP3(
        VmathSoaVector3 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	The constructed 3-D vector
pnt	3-D point

Return Values

None

Description

Construct a 3-D vector containing the x, y, and z elements of the specified 3-D point.

vmathSoaV3MakeFromScalar

Set all elements of a 3-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeFromScalar(
        VmathSoaVector3 *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed 3-D vector
scalar	Scalar value

Return Values

None

Description

Construct a 3-D vector with all elements set to the scalar value argument.

vmathSoaV3MakeXAxis

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeXAxis(
        VmathSoaVector3 *result
);
```

Arguments

result The constructed 3-D vector

Return Values

None

Description

Construct a 3-D vector equal to (1,0,0).

vmathSoaV3MakeYAxis

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeYAxis(
        VmathSoaVector3 *result
);
```

Arguments

`result` The constructed 3-D vector

Return Values

None

Description

Construct a 3-D vector equal to (0,1,0).

vmathSoaV3MakeZAxis

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MakeZAxis(
        VmathSoaVector3 *result
);
```

Arguments

result	The constructed 3-D vector

Return Values

None

Description

Construct a 3-D vector equal to (0,0,1).

vmathSoaV3MaxElem

Maximum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3MaxElem(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 3-D vector.

vmathSoaV3MaxPerElem

Maximum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MaxPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors.

vmathSoaV3MinElem

Minimum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3MinElem(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 3-D vector.

vmathSoaV3MinPerElem

Minimum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MinPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the minimum of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the minimum of the corresponding elements of two specified 3-D vectors.

vmathSoaV3MulPerElem

Multiply two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3MulPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	3-D vector in which each element is the product of the corresponding elements of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Multiply two 3-D vectors element by element.

vmathSoaV3Neg

Negate all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Neg(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	3-D vector containing negated elements of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Negate all elements of a 3-D vector.

vmathSoaV3Normalize

Normalize a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Normalize(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	The specified 3-D vector scaled to unit length
vec	3-D vector

Return Values

None

Description

Compute a normalized 3-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathSoaV3Outer

Outer product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Outer(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	The 3x3 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Compute the outer product of two 3-D vectors.

vmathSoaV3Print

Print a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Print(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

None

Description

Print a 3-D vector. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV3Prints

Print a 3-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Prints(
        const VmathSoaVector3 *vec,
        const char *name
);
```

Arguments

vec	3-D vector
name	String printed with the 3-D vector

Return Values

None

Description

Print a 3-D vector and an associated string identifier. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV3RecipPerElem

Compute the reciprocal of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3RecipPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathSoaV3RowMul

Pre-multiply a row vector by a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3RowMul(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	Product of a row-vector and a 3x3 matrix
vec	3-D vector
mat	3x3 matrix

Return Values

None

Description

Transpose a 3-D vector into a row vector and pre-multiply by 3x3 matrix.

vmathSoaV3RsqrtPerElem

Compute the reciprocal square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3RsqrtPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathSoaV3ScalarDiv

Divide a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3ScalarDiv(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec,
        vec_float4 scalar
);
```

Arguments

result	Quotient of the specified 3-D vector and scalar
vec	3-D vector
scalar	Scalar value

Return Values

None

Description

Divide a 3-D vector by a scalar.

vmathSoaV3ScalarMul

Multiply a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3ScalarMul(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec,
        vec_float4 scalar
);
```

Arguments

result	Product of the specified 3-D vector and scalar
vec	3-D vector
scalar	Scalar value

Return Values

None

Description

Multiply a 3-D vector by a scalar.

vmathSoaV3Select

Conditionally select between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Select(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the 3-D vector at the corresponding slot of *vec0* or *vec1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *vec0*, and a value of 0xFFFFFFFF selects the slot of *vec1*
vec0	3-D vector
vec1	3-D vector
select1	For each of the four word slots, this mask selects either the 3-D vector in the corresponding slot of *vec0* or the 3-D vector in the corresponding slot of *vec1*. A 0 bit selects from *vec0* whereas a 1 bit selects from *vec1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the 3-D vectors at each of the corresponding slots of *vec0* or *vec1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaV3SetElem

Set an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3SetElem(
        VmathSoaVector3 *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output 3-D vector
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathSoaV3SetX

Set the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3SetX(
        VmathSoaVector3 *result,
        vec_float4 x
);
```

Arguments

result	An output 3-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D vector to the specified scalar value.

vmathSoaV3SetY

Set the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3SetY(
        VmathSoaVector3 *result,
        vec_float4 y
);
```

Arguments

result	An output 3-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D vector to the specified scalar value.

vmathSoaV3SetZ

Set the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3SetZ(
        VmathSoaVector3 *result,
        vec_float4 z
);
```

Arguments

result	An output 3-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D vector to the specified scalar value.

vmathSoaV3Slerp

Spherical linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Slerp(
        VmathSoaVector3 *result,
        vec_float4 t,
        const VmathSoaVector3 *unitVec0,
        const VmathSoaVector3 *unitVec1
);
```

Arguments

result	Interpolated 3-D vector
t	Interpolation parameter
unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

None

Description

Perform spherical linear interpolation between two 3-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp `t` between 0 and 1.

vmathSoaV3SqrtPerElem

Compute the square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3SqrtPerElem(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector
vec	3-D vector

Return Values

None

Description

Create a 3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathSoaV3StoreHalfFloats

Store eight slots of two SoA 3-D vectors as half-floats.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3StoreHalfFloats(
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1,
        vec_ushort8 *threeQuads
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight slots of two SoA 3-D vectors in three quadwords of half-float values. Numbering slots of *vec0* as 0..3 and slots of *vec1* as 4..7, the output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathSoaV3StoreXYZArray

Store four slots of an SoA 3-D vector in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3StoreXYZArray(
        const VmathSoaVector3 *vec,
        vec_float4 *threeQuads
);
```

Arguments

vec	3-D vector
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four slots of an SoA 3-D vector in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathSoaV3Sub

Subtract a 3-D vector from another 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV3Sub(
        VmathSoaVector3 *result,
        const VmathSoaVector3 *vec0,
        const VmathSoaVector3 *vec1
);
```

Arguments

result	Difference of the specified 3-D vectors
vec0	3-D vector
vec1	3-D vector

Return Values

None

Description

Subtract a 3-D vector from another 3-D vector.

vmathSoaV3Sum

Compute the sum of all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV3Sum(
        const VmathSoaVector3 *vec
);
```

Arguments

vec 3-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 3-D vector.

4-D Vector Functions (SoA, by reference)

vmathSoaV4AbsPerElem

Compute the absolute value of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4AbsPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the absolute value of the corresponding element of vec
vec	4-D vector

Return Values

None

Description

Compute the absolute value of each element of a 4-D vector.

vmathSoaV4Add

Add two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Add(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	Sum of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Add two 4-D vectors.

vmathSoaV4Copy

Copy a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Copy(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	The constructed result
vec	4-D vector

Return Values

None

Description

Construct a copy of a 4-D vector.

vmathSoaV4CopySignPerElem

Copy sign from one 4-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4CopySignPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	4-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathSoaV4DivPerElem

Divide two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4DivPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the quotient of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Divide two 4-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathSoaV4Dot

Compute the dot product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4Dot(
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Dot product of the specified 4-D vectors

Description

Compute the dot product of two 4-D vectors.

vmathSoaV4Get4Aos

Extract four AoS 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Get4Aos(
        const VmathSoaVector4 *vec,
        VmathVector4 *result0,
        VmathVector4 *result1,
        VmathVector4 *result2,
        VmathVector4 *result3
);
```

Arguments

vec	4-D vector
result0	An output AoS 4-D vector
result1	An output AoS 4-D vector
result2	An output AoS 4-D vector
result3	An output AoS 4-D vector

Return Values

None

Description

Extract four AoS 4-D vectors from four slots of an SoA 4-D vector (transpose the data format).

vmathSoaV4GetElem

Get an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4GetElem(
        const VmathSoaVector4 *vec,
        int idx
);
```

Arguments

vec	4-D vector
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaV4GetW

Get the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4GetW(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

w element of a 4-D vector

Description

Get the w element of a 4-D vector.

vmathSoaV4GetX

Get the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4GetX(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

x element of a 4-D vector

Description

Get the x element of a 4-D vector.

vmathSoaV4GetXYZ

Get the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4GetXYZ(
        VmathSoaVector3 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result 3-D vector containing x, y, and z elements
vec 4-D vector

Return Values

None

Description

Extract a 4-D vector's x, y, and z elements into a 3-D vector.

vmathSoaV4GetY

Get the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4GetY(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

y element of a 4-D vector

Description

Get the y element of a 4-D vector.

vmathSoaV4GetZ

Get the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4GetZ(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

z element of a 4-D vector

Description

Get the z element of a 4-D vector.

vmathSoaV4Length

Compute the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4Length(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Length of the specified 4-D vector

Description

Compute the length of a 4-D vector.

vmathSoaV4LengthSqr

Compute the square of the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4LengthSqr(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Square of the length of the specified 4-D vector

Description

Compute the square of the length of a 4-D vector.

vmathSoaV4Lerp

Linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Lerp(
        VmathSoaVector4 *result,
        vec_float4 t,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	Interpolated 4-D vector
t	Interpolation parameter
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Linearly interpolate between two 4-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaV4MakeFrom4Aos

Insert four AoS 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFrom4Aos(
        VmathSoaVector4 *result,
        const VmathVector4 *vec0,
        const VmathVector4 *vec1,
        const VmathVector4 *vec2,
        const VmathVector4 *vec3
);
```

Arguments

result	The constructed SoA 4-D vector
vec0	AoS 4-D vector
vec1	AoS 4-D vector
vec2	AoS 4-D vector
vec3	AoS 4-D vector

Return Values

None

Description

Insert four AoS 4-D vectors into four slots of an SoA 4-D vector (transpose the data format).

vmathSoaV4MakeFromAos

Replicate an AoS 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromAos(
        VmathSoaVector4 *result,
        const VmathVector4 *vec
);
```

Arguments

result	The constructed SoA 4-D vector
vec	AoS 4-D vector

Return Values

None

Description

Replicate an AoS 4-D vector in all four slots of an SoA 4-D vector.

vmathSoaV4MakeFromElems

Construct a 4-D vector from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromElems(
        VmathSoaVector4 *result,
        vec_float4 x,
        vec_float4 y,
        vec_float4 z,
        vec_float4 w
);
```

Arguments

result	The 4-D vector that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value
w	Scalar value

Return Values

None

Description

Construct a 4-D vector containing the specified x, y, z, and w elements.

vmathSoaV4MakeFromP3

Copy x, y, and z from a 3-D point into a 4-D vector, and set w to 1.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromP3(
        VmathSoaVector4 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	The constructed 4-D vector
pnt	3-D point

Return Values

None

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D point and with the w element set to 1.

vmathSoaV4MakeFromQ

Copy elements from a quaternion into a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromQ(
        VmathSoaVector4 *result,
        const VmathSoaQuat *quat
);
```

Arguments

result	The constructed 4-D vector
quat	Quaternion

Return Values

None

Description

Construct a 4-D vector containing the x, y, z, and w elements of the specified quaternion.

vmathSoaV4MakeFromScalar

Set all elements of a 4-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromScalar(
        VmathSoaVector4 *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed 4-D vector
scalar	Scalar value

Return Values

None

Description

Construct a 4-D vector with all elements set to the scalar value argument.

vmathSoaV4MakeFromV3

Copy x, y, and z from a 3-D vector into a 4-D vector, and set w to 0.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromV3(
        VmathSoaVector4 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	The constructed 4-D vector
vec	3-D vector

Return Values

None

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to 0.

vmathSoaV4MakeFromV3Scalar

Construct a 4-D vector from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeFromV3Scalar(
        VmathSoaVector4 *result,
        const VmathSoaVector3 *xyz,
        vec_float4 w
);
```

Arguments

result	The constructed result
xyz	3-D vector
w	Scalar value

Return Values

None

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathSoaV4MakeWAxis

Construct w axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeWAxis(
        VmathSoaVector4 *result
);
```

Arguments

result The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (0,0,0,1).

vmathSoaV4MakeXAxis

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeXAxis(
        VmathSoaVector4 *result
);
```

Arguments

result	The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (1,0,0,0).

vmathSoaV4MakeYAxis

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeYAxis(
        VmathSoaVector4 *result
);
```

Arguments

result	The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (0,1,0,0).

vmathSoaV4MakeZAxis

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MakeZAxis(
        VmathSoaVector4 *result
);
```

Arguments

result The constructed 4-D vector

Return Values

None

Description

Construct a 4-D vector equal to (0,0,1,0).

vmathSoaV4MaxElem

Maximum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4MaxElem(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 4-D vector.

vmathSoaV4MaxPerElem

Maximum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MaxPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors.

vmathSoaV4MinElem

Minimum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4MinElem(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 4-D vector.

vmathSoaV4MinPerElem

Minimum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MinPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the minimum of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the minimum of the corresponding elements of two specified 4-D vectors.

vmathSoaV4MulPerElem

Multiply two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4MulPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	4-D vector in which each element is the product of the corresponding elements of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Multiply two 4-D vectors element by element.

vmathSoaV4Neg

Negate all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Neg(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	4-D vector containing negated elements of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Negate all elements of a 4-D vector.

vmathSoaV4Normalize

Normalize a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Normalize(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	The specified 4-D vector scaled to unit length
vec	4-D vector

Return Values

None

Description

Compute a normalized 4-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathSoaV4Outer

Outer product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Outer(
        VmathSoaMatrix4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	The 4x4 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Compute the outer product of two 4-D vectors.

vmathSoaV4Print

Print a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Print(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

None

Description

Print a 4-D vector. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV4Prints

Print a 4-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Prints(
        const VmathSoaVector4 *vec,
        const char *name
);
```

Arguments

vec	4-D vector
name	String printed with the 4-D vector

Return Values

None

Description

Print a 4-D vector and an associated string identifier. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV4RecipPerElem

Compute the reciprocal of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4RecipPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathSoaV4RsqrtPerElem

Compute the reciprocal square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4RsqrtPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathSoaV4ScalarDiv

Divide a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4ScalarDiv(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec,
        vec_float4 scalar
);
```

Arguments

result	Quotient of the specified 4-D vector and scalar
vec	4-D vector
scalar	Scalar value

Return Values

None

Description

Divide a 4-D vector by a scalar.

vmathSoaV4ScalarMul

Multiply a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4ScalarMul(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec,
        vec_float4 scalar
);
```

Arguments

result	Product of the specified 4-D vector and scalar
vec	4-D vector
scalar	Scalar value

Return Values

None

Description

Multiply a 4-D vector by a scalar.

vmathSoaV4Select

Conditionally select between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Select(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the 4-D vector at the corresponding slot of *vec0* or *vec1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *vec0*, and a value of 0xFFFFFFFF selects the slot of *vec1*
vec0	4-D vector
vec1	4-D vector
select1	For each of the four word slots, this mask selects either the 4-D vector in the corresponding slot of *vec0* or the 4-D vector in the corresponding slot of *vec1*. A 0 bit selects from *vec0* whereas a 1 bit selects from *vec1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the 4-D vectors at each of the corresponding slots of *vec0* or *vec1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaV4SetElem

Set an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SetElem(
        VmathSoaVector4 *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output 4-D vector
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaV4SetW

Set the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SetW(
        VmathSoaVector4 *result,
        vec_float4 w
);
```

Arguments

result	An output 4-D vector
w	Scalar value

Return Values

None

Description

Set the w element of a 4-D vector to the specified scalar value.

vmathSoaV4SetX

Set the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SetX(
        VmathSoaVector4 *result,
        vec_float4 x
);
```

Arguments

result	An output 4-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 4-D vector to the specified scalar value.

vmathSoaV4SetXYZ

Set the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SetXYZ(
        VmathSoaVector4 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	An output 4-D vector
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathSoaV4SetY

Set the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SetY(
        VmathSoaVector4 *result,
        vec_float4 y
);
```

Arguments

result	An output 4-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 4-D vector to the specified scalar value.

vmathSoaV4SetZ

Set the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SetZ(
        VmathSoaVector4 *result,
        vec_float4 z
);
```

Arguments

result	An output 4-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 4-D vector to the specified scalar value.

vmathSoaV4Slerp

Spherical linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Slerp(
        VmathSoaVector4 *result,
        vec_float4 t,
        const VmathSoaVector4 *unitVec0,
        const VmathSoaVector4 *unitVec1
);
```

Arguments

result	Interpolated 4-D vector
t	Interpolation parameter
unitVec0	4-D vector, expected to be unit-length
unitVec1	4-D vector, expected to be unit-length

Return Values

None

Description

Perform spherical linear interpolation between two 4-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp `t` between 0 and 1.

vmathSoaV4SqrtPerElem

Compute the square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4SqrtPerElem(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector
vec	4-D vector

Return Values

None

Description

Create a 4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathSoaV4StoreHalfFloats

Store four slots of an SoA 4-D vector as half-floats.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4StoreHalfFloats(
        const VmathSoaVector4 *vec,
        vec_ushort8 *twoQuads
);
```

Arguments

vec	4-D vector
twoQuads	An output array of 2 quadwords containing 16 half-floats

Return Values

None

Description

Store four slots of an SoA 4-D vector in two quadwords of half-float values. Numbering slots of *vec* as 0..3, the output is {x0,y0,z0,w0,x1,y1,z1,w1,x2,y2,z2,w2,x3,y3,z3,w3}.

vmathSoaV4Sub

Subtract a 4-D vector from another 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaV4Sub(
        VmathSoaVector4 *result,
        const VmathSoaVector4 *vec0,
        const VmathSoaVector4 *vec1
);
```

Arguments

result	Difference of the specified 4-D vectors
vec0	4-D vector
vec1	4-D vector

Return Values

None

Description

Subtract a 4-D vector from another 4-D vector.

vmathSoaV4Sum

Compute the sum of all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaV4Sum(
        const VmathSoaVector4 *vec
);
```

Arguments

vec 4-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 4-D vector.

Point Functions (SoA, by reference)

vmathSoaP3AbsPerElem

Compute the absolute value of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3AbsPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the absolute value of the corresponding element of pnt
pnt	3-D point

Return Values

None

Description

Compute the absolute value of each element of a 3-D point.

vmathSoaP3AddV3

Add a 3-D point to a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3AddV3(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt,
        const VmathSoaVector3 *vec
);
```

Arguments

result	Sum of the specified 3-D point and 3-D vector
pnt	3-D point
vec	3-D vector

Return Values

None

Description

Add a 3-D point to a 3-D vector.

vmathSoaP3Copy

Copy a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Copy(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	The constructed result
pnt	3-D point

Return Values

None

Description

Construct a copy of a 3-D point.

vmathSoaP3CopySignPerElem

Copy sign from one 3-D point to another, per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3CopySignPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element has the magnitude of the corresponding element of *pnt0* and the sign of the corresponding element of *pnt1*
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

For each element, create a value composed of the magnitude of *pnt0* and the sign of *pnt1*.

vmathSoaP3Dist

Compute the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3Dist(
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Distance between two 3-D points

Description

Compute the distance between two 3-D points.

vmathSoaP3DistFromOrigin

Compute the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3DistFromOrigin(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Distance of a 3-D point from the origin

Description

Compute the distance of a 3-D point from the coordinate-system origin.

vmathSoaP3DistSqr

Compute the square of the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3DistSqr(
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Square of the distance between two 3-D points

Description

Compute the square of the distance between two 3-D points.

vmathSoaP3DistSqrFromOrigin

Compute the square of the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3DistSqrFromOrigin(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Square of the distance of a 3-D point from the origin

Description

Compute the square of the distance of a 3-D point from the coordinate-system origin.

vmathSoaP3DivPerElem

Divide two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3DivPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the quotient of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Divide two 3-D points element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathSoaP3Get4Aos

Extract four AoS 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Get4Aos(
        const VmathSoaPoint3 *pnt,
        VmathPoint3 *result0,
        VmathPoint3 *result1,
        VmathPoint3 *result2,
        VmathPoint3 *result3
);
```

Arguments

pnt	3-D point
result0	An output AoS 3-D point
result1	An output AoS 3-D point
result2	An output AoS 3-D point
result3	An output AoS 3-D point

Return Values

None

Description

Extract four AoS 3-D points from four slots of an SoA 3-D point (transpose the data format).

vmathSoaP3GetElem

Get an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3GetElem(
        const VmathSoaPoint3 *pnt,
        int idx
);
```

Arguments

pnt	3-D point
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathSoaP3GetX

Get the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3GetX(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

x element of a 3-D point

Description

Get the x element of a 3-D point.

vmathSoaP3GetY

Get the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3GetY(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

y element of a 3-D point

Description

Get the y element of a 3-D point.

vmathSoaP3GetZ

Get the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3GetZ(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

z element of a 3-D point

Description

Get the z element of a 3-D point.

vmathSoaP3Lerp

Linear interpolation between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Lerp(
        VmathSoaPoint3 *result,
        vec_float4 t,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	Interpolated 3-D point
t	Interpolation parameter
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Linearly interpolate between two 3-D points.

Notes

Does not clamp `t` between 0 and 1.

vmathSoaP3LoadXYZArray

Load four three-float 3-D points, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3LoadXYZArray(
        VmathSoaPoint3 *pnt,
        const vec_float4 *threeQuads
);
```

Arguments

pnt	An output 3-D point
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D points, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four slots of an SoA 3-D point.

vmathSoaP3MakeFrom4Aos

Insert four AoS 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MakeFrom4Aos(
        VmathSoaPoint3 *result,
        const VmathPoint3 *pnt0,
        const VmathPoint3 *pnt1,
        const VmathPoint3 *pnt2,
        const VmathPoint3 *pnt3
);
```

Arguments

result	The constructed SoA 3-D point
pnt0	AoS 3-D point
pnt1	AoS 3-D point
pnt2	AoS 3-D point
pnt3	AoS 3-D point

Return Values

None

Description

Insert four AoS 3-D points into four slots of an SoA 3-D point (transpose the data format).

vmathSoaP3MakeFromAos

Replicate an AoS 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MakeFromAos(
        VmathSoaPoint3 *result,
        const VmathPoint3 *pnt
);
```

Arguments

result	The constructed SoA 3-D point
pnt	AoS 3-D point

Return Values

None

Description

Replicate an AoS 3-D point in all four slots of an SoA 3-D point.

vmathSoaP3MakeFromElems

Construct a 3-D point from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MakeFromElems(
        VmathSoaPoint3 *result,
        vec_float4 x,
        vec_float4 y,
        vec_float4 z
);
```

Arguments

result	The 3-D point that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value

Return Values

None

Description

Construct a 3-D point containing the specified x, y, and z elements.

vmathSoaP3MakeFromScalar

Set all elements of a 3-D point to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MakeFromScalar(
        VmathSoaPoint3 *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed 3-D point
scalar	Scalar value

Return Values

None

Description

Construct a 3-D point with all elements set to the scalar value argument.

vmathSoaP3MakeFromV3

Copy elements from a 3-D vector into a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MakeFromV3(
        VmathSoaPoint3 *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	The constructed 3-D point
vec	3-D vector

Return Values

None

Description

Construct a 3-D point containing the x, y, and z elements of the specified 3-D vector.

vmathSoaP3MaxElem

Maximum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3MaxElem(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Maximum value of all elements of *pnt*

Description

Compute the maximum value of all elements of a 3-D point.

vmathSoaP3MaxPerElem

Maximum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MaxPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points.

vmathSoaP3MinElem

Minimum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3MinElem(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Minimum value of all elements of *pnt*

Description

Compute the minimum value of all elements of a 3-D point.

vmathSoaP3MinPerElem

Minimum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MinPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the minimum of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the minimum of the corresponding elements of two specified 3-D points.

vmathSoaP3MulPerElem

Multiply two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3MulPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	3-D point in which each element is the product of the corresponding elements of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Multiply two 3-D points element by element.

vmathSoaP3NonUniformScale

Apply non-uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3NonUniformScale(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	3-D point in which each element is the product of the corresponding elements of the specified 3-D point and 3-D vector
pnt	3-D point
scaleVec	3-D vector

Return Values

None

Description

Apply non-uniform scale to a 3-D point.

vmathSoaP3Print

Print a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Print(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

None

Description

Print a 3-D point. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaP3Prints

Print a 3-D point and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Prints(
        const VmathSoaPoint3 *pnt,
        const char *name
);
```

Arguments

pnt	3-D point
name	String printed with the 3-D point

Return Values

None

Description

Print a 3-D point and an associated string identifier. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaP3Projection

Scalar projection of a 3-D point on a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3Projection(
        const VmathSoaPoint3 *pnt,
        const VmathSoaVector3 *unitVec
);
```

Arguments

pnt	3-D point
unitVec	3-D vector, expected to be unit-length

Return Values

Scalar projection of the 3-D point on the unit-length 3-D vector

Description

Scalar projection of a 3-D point on a unit-length 3-D vector (dot product).

vmathSoaP3RecipPerElem

Compute the reciprocal of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3RecipPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point
pnt	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function recipf4.

vmathSoaP3RsqrtPerElem

Compute the reciprocal square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3RsqrtPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point
pnt	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathSoaP3Scale

Apply uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Scale(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt,
        vec_float4 scaleVal
);
```

Arguments

result	3-D point in which every element is multiplied by the scalar value
pnt	3-D point
scaleVal	Scalar value

Return Values

None

Description

Apply uniform scale to a 3-D point.

vmathSoaP3Select

Conditionally select between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Select(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the 3-D point at the corresponding slot of *pnt0* or *pnt1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *pnt0*, and a value of 0xFFFFFFFF selects the slot of *pnt1*
pnt0	3-D point
pnt1	3-D point
select1	For each of the four word slots, this mask selects either the 3-D point in the corresponding slot of *pnt0* or the 3-D point in the corresponding slot of *pnt1*. A 0 bit selects from *pnt0* whereas a 1 bit selects from *pnt1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the 3-D points at each of the corresponding slots of *pnt0* or *pnt1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaP3SetElem

Set an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3SetElem(
        VmathSoaPoint3 *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output 3-D point
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathSoaP3SetX

Set the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3SetX(
        VmathSoaPoint3 *result,
        vec_float4 x
);
```

Arguments

result	An output 3-D point
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D point to the specified scalar value.

vmathSoaP3SetY

Set the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3SetY(
        VmathSoaPoint3 *result,
        vec_float4 y
);
```

Arguments

result	An output 3-D point
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D point to the specified scalar value.

vmathSoaP3SetZ

Set the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3SetZ(
        VmathSoaPoint3 *result,
        vec_float4 z
);
```

Arguments

result	An output 3-D point
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D point to the specified scalar value.

vmathSoaP3SqrtPerElem

Compute the square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3SqrtPerElem(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	3-D point in which each element is the square root of the corresponding element of the specified 3-D point
pnt	3-D point

Return Values

None

Description

Create a 3-D point in which each element is the square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathSoaP3StoreHalfFloats

Store eight slots of two SoA 3-D points as half-floats.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3StoreHalfFloats(
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1,
        vec_ushort8 *threeQuads
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight slots of two SoA 3-D points in three quadwords of half-float values. Numbering slots of *pnt0* as 0..3 and slots of *pnt1* as 4..7, the output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathSoaP3StoreXYZArray

Store four slots of an SoA 3-D point in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3StoreXYZArray(
        const VmathSoaPoint3 *pnt,
        vec_float4 *threeQuads
);
```

Arguments

pnt	3-D point
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four slots of an SoA 3-D point in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathSoaP3Sub

Subtract a 3-D point from another 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3Sub(
        VmathSoaVector3 *result,
        const VmathSoaPoint3 *pnt0,
        const VmathSoaPoint3 *pnt1
);
```

Arguments

result	Difference of the specified 3-D points
pnt0	3-D point
pnt1	3-D point

Return Values

None

Description

Subtract a 3-D point from another 3-D point.

vmathSoaP3SubV3

Subtract a 3-D vector from a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaP3SubV3(
        VmathSoaPoint3 *result,
        const VmathSoaPoint3 *pnt,
        const VmathSoaVector3 *vec
);
```

Arguments

result	Difference of the specified 3-D point and 3-D vector
pnt	3-D point
vec	3-D vector

Return Values

None

Description

Subtract a 3-D vector from a 3-D point.

vmathSoaP3Sum

Compute the sum of all elements of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaP3Sum(
        const VmathSoaPoint3 *pnt
);
```

Arguments

pnt 3-D point

Return Values

Sum of all elements of *pnt*

Description

Compute the sum of all elements of a 3-D point.

Quaternion Functions (SoA, by reference)

vmathSoaQAdd

Add two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQAdd(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat0,
        const VmathSoaQuat *quat1
);
```

Arguments

result	Sum of the specified quaternions
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Add two quaternions.

vmathSoaQConj

Compute the conjugate of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQConj(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat
);
```

Arguments

result	Conjugate of the specified quaternion
quat	Quaternion

Return Values

None

Description

Compute the conjugate of a quaternion.

vmathSoaQCopy

Copy a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQCopy(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat
);
```

Arguments

result	The constructed result
quat	Quaternion

Return Values

None

Description

Construct a copy of a quaternion.

vmathSoaQDot

Compute the dot product of two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQDot(
        const VmathSoaQuat *quat0,
        const VmathSoaQuat *quat1
);
```

Arguments

quat0 Quaternion
quat1 Quaternion

Return Values

Dot product of the specified quaternions

Description

Compute the dot product of two quaternions.

vmathSoaQGet4Aos

Extract four AoS quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQGet4Aos(
        const VmathSoaQuat *quat,
        VmathQuat *result0,
        VmathQuat *result1,
        VmathQuat *result2,
        VmathQuat *result3
);
```

Arguments

quat	Quaternion
result0	An output AoS quaternion
result1	An output AoS quaternion
result2	An output AoS quaternion
result3	An output AoS quaternion

Return Values

None

Description

Extract four AoS quaternions from four slots of an SoA quaternion (transpose the data format).

vmathSoaQGetElem

Get an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQGetElem(
        const VmathSoaQuat *quat,
        int idx
);
```

Arguments

quat	Quaternion
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaQGetW

Get the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQGetW(
        const VmathSoaQuat *quat
);
```

Arguments

quat Quaternion

Return Values

w element of a quaternion

Description

Get the w element of a quaternion.

vmathSoaQGetX

Get the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQGetX(
        const VmathSoaQuat *quat
);
```

Arguments

quat Quaternion

Return Values

x element of a quaternion

Description

Get the x element of a quaternion.

vmathSoaQGetXYZ

Get the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQGetXYZ(
        VmathSoaVector3 *result,
        const VmathSoaQuat *quat
);
```

Arguments

result	3-D vector containing x, y, and z elements
quat	Quaternion

Return Values

None

Description

Extract a quaternion's x, y, and z elements into a 3-D vector.

vmathSoaQGetY

Get the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQGetY(
        const VmathSoaQuat *quat
);
```

Arguments

quat Quaternion

Return Values

y element of a quaternion

Description

Get the y element of a quaternion.

vmathSoaQGetZ

Get the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQGetZ(
        const VmathSoaQuat *quat
);
```

Arguments

quat Quaternion

Return Values

z element of a quaternion

Description

Get the z element of a quaternion.

vmathSoaQLength

Compute the length of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQLength(
        const VmathSoaQuat *quat
);
```

Arguments

quat	Quaternion

Return Values

Length of the specified quaternion

Description

Compute the length of a quaternion.

vmathSoaQLerp

Linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQLerp(
        VmathSoaQuat *result,
        vec_float4 t,
        const VmathSoaQuat *quat0,
        const VmathSoaQuat *quat1
);
```

Arguments

result	Interpolated quaternion
t	Interpolation parameter
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Linearly interpolate between two quaternions.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaQMakeFrom4Aos

Insert four AoS quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFrom4Aos(
        VmathSoaQuat *result,
        const VmathQuat *quat0,
        const VmathQuat *quat1,
        const VmathQuat *quat2,
        const VmathQuat *quat3
);
```

Arguments

result	The constructed SoA quaternion
quat0	AoS quaternion
quat1	AoS quaternion
quat2	AoS quaternion
quat3	AoS quaternion

Return Values

None

Description

Insert four AoS quaternions into four slots of an SoA quaternion (transpose the data format).

vmathSoaQMakeFromAos

Replicate an AoS quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFromAos(
        VmathSoaQuat *result,
        const VmathQuat *quat
);
```

Arguments

result	The constructed SoA quaternion
quat	AoS quaternion

Return Values

None

Description

Replicate an AoS quaternion in all four slots of an SoA quaternion.

vmathSoaQMakeFromElems

Construct a quaternion from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFromElems(
        VmathSoaQuat *result,
        vec_float4 x,
        vec_float4 y,
        vec_float4 z,
        vec_float4 w
);
```

Arguments

result	The quaternion that contains the specified elements
x	Scalar value
y	Scalar value
z	Scalar value
w	Scalar value

Return Values

None

Description

Construct a quaternion containing the specified x, y, z, and w elements.

vmathSoaQMakeFromM3

Convert a rotation matrix to a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFromM3(
        VmathSoaQuat *result,
        const VmathSoaMatrix3 *rotMat
);
```

Arguments

result	The constructed result
rotMat	3x3 matrix, expected to be a rotation matrix

Return Values

None

Description

Construct a unit-length quaternion representing the same transformation as a rotation matrix.

vmathSoaQMakeFromScalar

Set all elements of a quaternion to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFromScalar(
        VmathSoaQuat *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed quaternion
scalar	Scalar value

Return Values

None

Description

Construct a quaternion with all elements set to the scalar value argument.

vmathSoaQMakeFromV3Scalar

Construct a quaternion from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFromV3Scalar(
        VmathSoaQuat *result,
        const VmathSoaVector3 *xyz,
        vec_float4 w
);
```

Arguments

result	The constructed result
xyz	3-D vector
w	Scalar value

Return Values

None

Description

Construct a quaternion with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathSoaQMakeFromV4

Copy elements from a 4-D vector into a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeFromV4(
        VmathSoaQuat *result,
        const VmathSoaVector4 *vec
);
```

Arguments

result	The constructed quaternion
vec	4-D vector

Return Values

None

Description

Construct a quaternion containing the x, y, z, and w elements of the specified 4-D vector.

vmathSoaQMakeIdentity

Construct an identity quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeIdentity(
        VmathSoaQuat *result
);
```

Arguments

result The constructed quaternion

Return Values

None

Description

Construct an identity quaternion equal to (0,0,0,1).

vmathSoaQMakeRotationArc

Construct a quaternion to rotate between two unit-length 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeRotationArc(
        VmathSoaQuat *result,
        const VmathSoaVector3 *unitVec0,
        const VmathSoaVector3 *unitVec1
);
```

Arguments

result	The constructed quaternion
unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a quaternion to rotate between two unit-length 3-D vectors.

Notes

The result is unpredictable if *unitVec0* and *unitVec1* point in opposite directions.

vmathSoaQMakeRotationAxis

Construct a quaternion to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeRotationAxis(
        VmathSoaQuat *result,
        vec_float4 radians,
        const VmathSoaVector3 *unitVec
);
```

Arguments

result	The constructed quaternion
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a quaternion to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaQMakeRotationX

Construct a quaternion to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeRotationX(
        VmathSoaQuat *result,
        vec_float4 radians
);
```

Arguments

result	The constructed quaternion
radians	Scalar value

Return Values

None

Description

Construct a quaternion to rotate around the x axis by the specified radians angle.

vmathSoaQMakeRotationY

Construct a quaternion to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeRotationY(
        VmathSoaQuat *result,
        vec_float4 radians
);
```

Arguments

result	The constructed quaternion
radians	Scalar value

Return Values

None

Description

Construct a quaternion to rotate around the y axis by the specified radians angle.

vmathSoaQMakeRotationZ

Construct a quaternion to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMakeRotationZ(
        VmathSoaQuat *result,
        vec_float4 radians
);
```

Arguments

result	The constructed quaternion
radians	Scalar value

Return Values

None

Description

Construct a quaternion to rotate around the z axis by the specified radians angle.

vmathSoaQMul

Multiply two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQMul(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat0,
        const VmathSoaQuat *quat1
);
```

Arguments

result	Product of the specified quaternions
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Multiply two quaternions.

vmathSoaQNeg

Negate all elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQNeg(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat
);
```

Arguments

result	Quaternion containing negated elements of the specified quaternion
quat	Quaternion

Return Values

None

Description

Negate all elements of a quaternion.

vmathSoaQNorm

Compute the norm of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaQNorm(
        const VmathSoaQuat *quat
);
```

Arguments

quat Quaternion

Return Values

The norm of the specified quaternion

Description

Compute the norm, equal to the square of the length, of a quaternion.

vmathSoaQNormalize

Normalize a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQNormalize(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat
);
```

Arguments

result	The specified quaternion scaled to unit length
quat	Quaternion

Return Values

None

Description

Compute a normalized quaternion.

Notes

The result is unpredictable when all elements of quat are at or near zero.

vmathSoaQPrint

Print a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQPrint(
        const VmathSoaQuat *quat
);
```

Arguments

quat	Quaternion

Return Values

None

Description

Print a quaternion.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaQPrints

Print a quaternion and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQPrints(
        const VmathSoaQuat *quat,
        const char *name
);
```

Arguments

quat	Quaternion
name	String printed with the quaternion

Return Values

None

Description

Print a quaternion and an associated string identifier.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaQRotate

Use a unit-length quaternion to rotate a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQRotate(
        VmathSoaVector3 *result,
        const VmathSoaQuat *unitQuat,
        const VmathSoaVector3 *vec
);
```

Arguments

result	The rotated 3-D vector, equivalent to unitQuat*Quat(vec,0)*conj(unitQuat)
unitQuat	Quaternion, expected to be unit-length
vec	3-D vector

Return Values

None

Description

Rotate a 3-D vector by applying a unit-length quaternion.

vmathSoaQScalarDiv

Divide a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQScalarDiv(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat,
        vec_float4 scalar
);
```

Arguments

result	Quotient of the specified quaternion and scalar
quat	Quaternion
scalar	Scalar value

Return Values

None

Description

Divide a quaternion by a scalar.

vmathSoaQScalarMul

Multiply a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQScalarMul(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat,
        vec_float4 scalar
);
```

Arguments

result	Product of the specified quaternion and scalar
quat	Quaternion
scalar	Scalar value

Return Values

None

Description

Multiply a quaternion by a scalar.

vmathSoaQSelect

Conditionally select between two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSelect(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat0,
        const VmathSoaQuat *quat1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the quaternion at the corresponding slot of *quat0* or *quat1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *quat0*, and a value of 0xFFFFFFFF selects the slot of *quat1*
quat0	Quaternion
quat1	Quaternion
select1	For each of the four word slots, this mask selects either the quaternion in the corresponding slot of *quat0* or the quaternion in the corresponding slot of *quat1*. A 0 bit selects from *quat0* whereas a 1 bit selects from *quat1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the quaternions at each of the corresponding slots of *quat0* or *quat1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaQSetElem

Set an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSetElem(
        VmathSoaQuat *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output quaternion
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaQSetW

Set the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSetW(
        VmathSoaQuat *result,
        vec_float4 w
);
```

Arguments

result	An output quaternion
w	Scalar value

Return Values

None

Description

Set the w element of a quaternion to the specified scalar value.

vmathSoaQSetX

Set the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSetX(
        VmathSoaQuat *result,
        vec_float4 x
);
```

Arguments

result	An output quaternion
x	Scalar value

Return Values

None

Description

Set the x element of a quaternion to the specified scalar value.

vmathSoaQSetXYZ

Set the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSetXYZ(
        VmathSoaQuat *result,
        const VmathSoaVector3 *vec
);
```

Arguments

result	An output quaternion
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathSoaQSetY

Set the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSetY(
        VmathSoaQuat *result,
        vec_float4 y
);
```

Arguments

result	An output quaternion
y	Scalar value

Return Values

None

Description

Set the y element of a quaternion to the specified scalar value.

vmathSoaQSetZ

Set the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSetZ(
        VmathSoaQuat *result,
        vec_float4 z
);
```

Arguments

result	An output quaternion
z	Scalar value

Return Values

None

Description

Set the z element of a quaternion to the specified scalar value.

vmathSoaQSlerp

Spherical linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSlerp(
        VmathSoaQuat *result,
        vec_float4 t,
        const VmathSoaQuat *unitQuat0,
        const VmathSoaQuat *unitQuat1
);
```

Arguments

result	Interpolated quaternion
t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length

Return Values

None

Description

Perform spherical linear interpolation between two quaternions.

Notes

Interpolates along the shortest path between orientations. Does not clamp `t` between 0 and 1.

vmathSoaQSquad

Spherical quadrangle interpolation.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSquad(
        VmathSoaQuat *result,
        vec_float4 t,
        const VmathSoaQuat *unitQuat0,
        const VmathSoaQuat *unitQuat1,
        const VmathSoaQuat *unitQuat2,
        const VmathSoaQuat *unitQuat3
);
```

Arguments

result	Interpolated quaternion
t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length
unitQuat2	Quaternion, expected to be unit-length
unitQuat3	Quaternion, expected to be unit-length

Return Values

None

Description

Perform spherical quadrangle interpolation between four quaternions.

vmathSoaQSub

Subtract a quaternion from another quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaQSub(
        VmathSoaQuat *result,
        const VmathSoaQuat *quat0,
        const VmathSoaQuat *quat1
);
```

Arguments

result	Difference of the specified quaternions
quat0	Quaternion
quat1	Quaternion

Return Values

None

Description

Subtract a quaternion from another quaternion.

3x3 Matrix Functions (SoA, by reference)

vmathSoaM3AbsPerElem

Compute the absolute value of a 3x3 matrix per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3AbsPerElem(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	3x3 matrix in which each element is the absolute value of the corresponding element of the specified 3x3 matrix
mat	3x3 matrix

Return Values

None

Description

Compute the absolute value of each element of a 3x3 matrix.

vmathSoaM3Add

Add two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Add(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat0,
        const VmathSoaMatrix3 *mat1
);
```

Arguments

result	Sum of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Add two 3x3 matrices.

vmathSoaM3AppendScale

Append (post-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3AppendScale(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	The product of *mat* and a scale transformation created from *scaleVec*
mat	3x3 matrix
scaleVec	3-D vector

Return Values

None

Description

Post-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM3Copy

Copy a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Copy(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	The constructed result
mat	3x3 matrix

Return Values

None

Description

Construct a copy of a 3x3 matrix.

vmathSoaM3Determinant

Determinant of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaM3Determinant(
        const VmathSoaMatrix3 *mat
);
```

Arguments

mat 3x3 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 3x3 matrix.

vmathSoaM3Get4Aos

Extract four AoS 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Get4Aos(
        const VmathSoaMatrix3 *mat,
        VmathMatrix3 *result0,
        VmathMatrix3 *result1,
        VmathMatrix3 *result2,
        VmathMatrix3 *result3
);
```

Arguments

mat	3x3 matrix
result0	An output AoS 3x3 matrix
result1	An output AoS 3x3 matrix
result2	An output AoS 3x3 matrix
result3	An output AoS 3x3 matrix

Return Values

None

Description

Extract four AoS 3x3 matrices from four slots of an SoA 3x3 matrix (transpose the data format).

vmathSoaM3GetCol

Get the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3GetCol(
        VmathSoaVector3 *result,
        const VmathSoaMatrix3 *mat,
        int col
);
```

Arguments

result	The column referred to by the specified index
mat	3x3 matrix
col	Index, expected in the range 0-2

Return Values

None

Description

Get the column of a 3x3 matrix referred to by the specified index.

vmathSoaM3GetCol0

Get column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3GetCol0(
        VmathSoaVector3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	Column 0
mat	3x3 matrix

Return Values

None

Description

Get column 0 of a 3x3 matrix.

vmathSoaM3GetCol1

Get column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3GetCol1(
        VmathSoaVector3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	Column 1
mat	3x3 matrix

Return Values

None

Description

Get column 1 of a 3x3 matrix.

vmathSoaM3GetCol2

Get column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3GetCol2(
        VmathSoaVector3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	Column 2
mat	3x3 matrix

Return Values

None

Description

Get column 2 of a 3x3 matrix.

vmathSoaM3GetElem

Get the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaM3GetElem(
        const VmathSoaMatrix3 *mat,
        int col,
        int row
);
```

Arguments

mat	3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x3 matrix referred to by column and row indices.

vmathSoaM3GetRow

Get the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3GetRow(
        VmathSoaVector3 *result,
        const VmathSoaMatrix3 *mat,
        int row
);
```

Arguments

result	The row referred to by the specified index
mat	3x3 matrix
row	Index, expected in the range 0-2

Return Values

None

Description

Get the row of a 3x3 matrix referred to by the specified index.

vmathSoaM3Inverse

Compute the inverse of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Inverse(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	Inverse of *mat*
mat	3x3 matrix

Return Values

None

Description

Compute the inverse of a 3x3 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathSoaM3MakeFrom4Aos

Insert four AoS 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeFrom4Aos(
        VmathSoaMatrix3 *result,
        const VmathMatrix3 *mat0,
        const VmathMatrix3 *mat1,
        const VmathMatrix3 *mat2,
        const VmathMatrix3 *mat3
);
```

Arguments

result	The constructed 3x3 matrix
mat0	AoS 3x3 matrix
mat1	AoS 3x3 matrix
mat2	AoS 3x3 matrix
mat3	AoS 3x3 matrix

Return Values

None

Description

Insert four AoS 3x3 matrices into four slots of an SoA 3x3 matrix (transpose the data format).

vmathSoaM3MakeFromAos

Replicate an AoS 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeFromAos(
        VmathSoaMatrix3 *result,
        const VmathMatrix3 *mat
);
```

Arguments

result	The constructed 3x3 matrix
mat	AoS 3x3 matrix

Return Values

None

Description

Replicate an AoS 3x3 matrix in all four slots of an SoA 3x3 matrix.

vmathSoaM3MakeFromCols

Construct a 3x3 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeFromCols(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *col0,
        const VmathSoaVector3 *col1,
        const VmathSoaVector3 *col2
);
```

Arguments

result	The 3x3 matrix that contains the specified columns
col0	3-D vector
col1	3-D vector
col2	3-D vector

Return Values

None

Description

Construct a 3x3 matrix containing the specified columns.

vmathSoaM3MakeFromQ

Construct a 3x3 rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeFromQ(
        VmathSoaMatrix3 *result,
        const VmathSoaQuat *unitQuat
);
```

Arguments

result	A 3x3 matrix that applies the same rotation as *unitQuat*
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaM3MakeFromScalar

Set all elements of a 3x3 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeFromScalar(
        VmathSoaMatrix3 *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed 3x3 matrix
scalar	Scalar value

Return Values

None

Description

Construct a 3x3 matrix with all elements set to the scalar value argument.

vmathSoaM3MakeIdentity

Construct an identity 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeIdentity(
        VmathSoaMatrix3 *result
);
```

Arguments

result	The constructed 3x3 matrix

Return Values

None

Description

Construct an identity 3x3 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathSoaM3MakeRotationAxis

Construct a 3x3 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeRotationAxis(
        VmathSoaMatrix3 *result,
        vec_float4 radians,
        const VmathSoaVector3 *unitVec
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a 3x3 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaM3MakeRotationQ

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeRotationQ(
        VmathSoaMatrix3 *result,
        const VmathSoaQuat *unitQuat
);
```

Arguments

result	A 3x3 matrix that applies the same rotation as `unitQuat`
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaM3MakeRotationX

Construct a 3x3 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeRotationX(
        VmathSoaMatrix3 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x3 matrix to rotate around the x axis by the specified radians angle.

vmathSoaM3MakeRotationY

Construct a 3x3 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeRotationY(
        VmathSoaMatrix3 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x3 matrix to rotate around the y axis by the specified radians angle.

vmathSoaM3MakeRotationZ

Construct a 3x3 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeRotationZ(
        VmathSoaMatrix3 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 3x3 matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x3 matrix to rotate around the z axis by the specified radians angle.

vmathSoaM3MakeRotationZYX

Construct a 3x3 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeRotationZYX(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *radiansXYZ
);
```

Arguments

result	The constructed 3x3 matrix
radiansXYZ	3-D vector

Return Values

None

Description

Construct a 3x3 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to `rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())`.

vmathSoaM3MakeScale

Construct a 3x3 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MakeScale(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	The constructed 3x3 matrix
scaleVec	3-D vector

Return Values

None

Description

Construct a 3x3 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathSoaM3Mul

Multiply two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Mul(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat0,
        const VmathSoaMatrix3 *mat1
);
```

Arguments

result	Product of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Multiply two 3x3 matrices.

vmathSoaM3MulPerElem

Multiply two 3x3 matrices per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MulPerElem(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat0,
        const VmathSoaMatrix3 *mat1
);
```

Arguments

result	3x3 matrix in which each element is the product of the corresponding elements of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Multiply two 3x3 matrices element by element.

vmathSoaM3MulV3

Multiply a 3x3 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3MulV3(
        VmathSoaVector3 *result,
        const VmathSoaMatrix3 *mat,
        const VmathSoaVector3 *vec
);
```

Arguments

result	Product of the specified 3x3 matrix and 3-D vector
mat	3x3 matrix
vec	3-D vector

Return Values

None

Description

Multiply a 3x3 matrix by a 3-D vector.

vmathSoaM3Neg

Negate all elements of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Neg(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	3x3 matrix containing negated elements of the specified 3x3 matrix
mat	3x3 matrix

Return Values

None

Description

Negate all elements of a 3x3 matrix.

vmathSoaM3PrependScale

Prepend (pre-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3PrependScale(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *scaleVec,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	The product of a scale transformation created from *scaleVec* and *mat*
scaleVec	3-D vector
mat	3x3 matrix

Return Values

None

Description

Pre-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM3Print

Print a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Print(
        const VmathSoaMatrix3 *mat
);
```

Arguments

mat 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM3Prints

Print a 3x3 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Prints(
        const VmathSoaMatrix3 *mat,
        const char *name
);
```

Arguments

mat	3x3 matrix
name	String printed with the 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix and an associated string identifier. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM3ScalarMul

Multiply a 3x3 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3ScalarMul(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat,
        vec_float4 scalar
);
```

Arguments

result	Product of the specified 3x3 matrix and scalar
mat	3x3 matrix
scalar	Scalar value

Return Values

None

Description

Multiply a 3x3 matrix by a scalar.

vmathSoaM3Select

Conditionally select between two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Select(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat0,
        const VmathSoaMatrix3 *mat1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the 3x3 matrix at the corresponding slot of *mat0* or *mat1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *mat0* and a value of 0xFFFFFFFF selects the slot of *mat1*
mat0	3x3 matrix
mat1	3x3 matrix
select1	For each of the four word slots, this mask selects either the 3x3 matrix in the corresponding slot of *mat0* or the 3x3 matrix in the corresponding slot of *mat1*. A 0 bit selects from *mat0* whereas a 1 bit selects from *mat1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the 3x3 matrices at each of the corresponding slots of *mat0* or *mat1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaM3SetCol

Set the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3SetCol(
        VmathSoaMatrix3 *result,
        int col,
        const VmathSoaVector3 *vec
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the column of a 3x3 matrix referred to by the specified index.

vmathSoaM3SetCol0

Set column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3SetCol0(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *col0
);
```

Arguments

result	An output 3x3 matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x3 matrix.

vmathSoaM3SetCol1

Set column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3SetCol1(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *col1
);
```

Arguments

result	An output 3x3 matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x3 matrix.

vmathSoaM3SetCol2

Set column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3SetCol2(
        VmathSoaMatrix3 *result,
        const VmathSoaVector3 *col2
);
```

Arguments

result	An output 3x3 matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x3 matrix.

vmathSoaM3SetElem

Set the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3SetElem(
        VmathSoaMatrix3 *result,
        int col,
        int row,
        vec_float4 val
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x3 matrix referred to by column and row indices.

vmathSoaM3SetRow

Set the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3SetRow(
        VmathSoaMatrix3 *result,
        int row,
        const VmathSoaVector3 *vec
);
```

Arguments

result	An output 3x3 matrix
row	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the row of a 3x3 matrix referred to by the specified index.

vmathSoaM3Sub

Subtract a 3x3 matrix from another 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Sub(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat0,
        const VmathSoaMatrix3 *mat1
);
```

Arguments

result	Difference of the specified 3x3 matrices
mat0	3x3 matrix
mat1	3x3 matrix

Return Values

None

Description

Subtract a 3x3 matrix from another 3x3 matrix.

vmathSoaM3Transpose

Transpose of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM3Transpose(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix3 *mat
);
```

Arguments

result	*mat* transposed
mat	3x3 matrix

Return Values

None

Description

Compute the transpose of a 3x3 matrix.

4x4 Matrix Functions (SoA, by reference)

vmathSoaM4AbsPerElem

Compute the absolute value of a 4x4 matrix per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4AbsPerElem(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	4x4 matrix in which each element is the absolute value of the corresponding element of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Compute the absolute value of each element of a 4x4 matrix.

vmathSoaM4Add

Add two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Add(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat0,
        const VmathSoaMatrix4 *mat1
);
```

Arguments

result	Sum of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Add two 4x4 matrices.

vmathSoaM4AffineInverse

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4AffineInverse(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Inverse of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is inverse(M), whose translation vector is -inverse(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions. The result is unpredictable when the determinant of `mat` is equal to or near 0.

vmathSoaM4AppendScale

Append (post-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4AppendScale(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	The product of *mat* and a scale transformation created from *scaleVec*
mat	4x4 matrix
scaleVec	3-D vector

Return Values

None

Description

Post-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM4Copy

Copy a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Copy(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	The constructed result
mat	4x4 matrix

Return Values

None

Description

Construct a copy of a 4x4 matrix.

vmathSoaM4Determinant

Determinant of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaM4Determinant(
        const VmathSoaMatrix4 *mat
);
```

Arguments

mat 4x4 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 4x4 matrix.

vmathSoaM4Get4Aos

Extract four AoS 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Get4Aos(
        const VmathSoaMatrix4 *mat,
        VmathMatrix4 *result0,
        VmathMatrix4 *result1,
        VmathMatrix4 *result2,
        VmathMatrix4 *result3
);
```

Arguments

mat	4x4 matrix
result0	An output AoS 4x4 matrix
result1	An output AoS 4x4 matrix
result2	An output AoS 4x4 matrix
result3	An output AoS 4x4 matrix

Return Values

None

Description

Extract four AoS 4x4 matrices from four slots of an SoA 4x4 matrix (transpose the data format).

vmathSoaM4GetCol

Get the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetCol(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat,
        int col
);
```

Arguments

result	The column referred to by the specified index
mat	4x4 matrix
col	Index, expected in the range 0-3

Return Values

None

Description

Get the column of a 4x4 matrix referred to by the specified index.

vmathSoaM4GetCol0

Get column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetCol0(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Column 0
mat	4x4 matrix

Return Values

None

Description

Get column 0 of a 4x4 matrix.

vmathSoaM4GetCol1

Get column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetCol1(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Column 1
mat	4x4 matrix

Return Values

None

Description

Get column 1 of a 4x4 matrix.

vmathSoaM4GetCol2

Get column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetCol2(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Column 2
mat	4x4 matrix

Return Values

None

Description

Get column 2 of a 4x4 matrix.

vmathSoaM4GetCol3

Get column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetCol3(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Column 3
mat	4x4 matrix

Return Values

None

Description

Get column 3 of a 4x4 matrix.

vmathSoaM4GetElem

Get the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaM4GetElem(
        const VmathSoaMatrix4 *mat,
        int col,
        int row
);
```

Arguments

mat	4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3

Return Values

Element selected by *col* and *row*

Description

Get the element of a 4x4 matrix referred to by column and row indices.

vmathSoaM4GetRow

Get the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetRow(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat,
        int row
);
```

Arguments

result	The row referred to by the specified index
mat	4x4 matrix
row	Index, expected in the range 0-3

Return Values

None

Description

Get the row of a 4x4 matrix referred to by the specified index.

vmathSoaM4GetTranslation

Get the translation component of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetTranslation(
        VmathSoaVector3 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Translation component
mat	4x4 matrix

Return Values

None

Description

Get the translation component of a 4x4 matrix.

vmathSoaM4GetUpper3x3

Get the upper-left 3x3 submatrix of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4GetUpper3x3(
        VmathSoaMatrix3 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Upper-left 3x3 submatrix
mat	4x4 matrix

Return Values

None

Description

Get the upper-left 3x3 submatrix of a 4x4 matrix.

vmathSoaM4Inverse

Compute the inverse of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Inverse(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Inverse of *mat*
mat	4x4 matrix

Return Values

None

Description

Compute the inverse of a 4x4 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathSoaM4MakeFrom4Aos

Insert four AoS 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFrom4Aos(
        VmathSoaMatrix4 *result,
        const VmathMatrix4 *mat0,
        const VmathMatrix4 *mat1,
        const VmathMatrix4 *mat2,
        const VmathMatrix4 *mat3
);
```

Arguments

result	The constructed 4x4 matrix
mat0	AoS 4x4 matrix
mat1	AoS 4x4 matrix
mat2	AoS 4x4 matrix
mat3	AoS 4x4 matrix

Return Values

None

Description

Insert four AoS 4x4 matrices into four slots of an SoA 4x4 matrix (transpose the data format).

vmathSoaM4MakeFromAos

Replicate an AoS 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFromAos(
        VmathSoaMatrix4 *result,
        const VmathMatrix4 *mat
);
```

Arguments

result	The constructed 4x4 matrix
mat	AoS 4x4 matrix

Return Values

None

Description

Replicate an AoS 4x4 matrix in all four slots of an SoA 4x4 matrix.

vmathSoaM4MakeFromCols

Construct a 4x4 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFromCols(
        VmathSoaMatrix4 *result,
        const VmathSoaVector4 *col0,
        const VmathSoaVector4 *col1,
        const VmathSoaVector4 *col2,
        const VmathSoaVector4 *col3
);
```

Arguments

result	The 4x4 matrix that contains the specified columns
col0	4-D vector
col1	4-D vector
col2	4-D vector
col3	4-D vector

Return Values

None

Description

Construct a 4x4 matrix containing the specified columns.

vmathSoaM4MakeFromM3V3

Construct a 4x4 matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFromM3V3(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix3 *mat,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	The constructed 4x4 matrix
mat	3x3 matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix whose upper 3x3 elements are equal to the 3x3 matrix argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathSoaM4MakeFromQV3

Construct a 4x4 matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFromQV3(
        VmathSoaMatrix4 *result,
        const VmathSoaQuat *unitQuat,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	The constructed 4x4 matrix
unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathSoaM4MakeFromScalar

Set all elements of a 4x4 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFromScalar(
        VmathSoaMatrix4 *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed 4x4 matrix
scalar	Scalar value

Return Values

None

Description

Construct a 4x4 matrix with all elements set to the scalar value argument.

vmathSoaM4MakeFromT3

Construct a 4x4 matrix from a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFromT3(
        VmathSoaMatrix4 *result,
        const VmathSoaTransform3 *mat
);
```

Arguments

result	The constructed 4x4 matrix
mat	3x4 transformation matrix

Return Values

None

Description

Construct a 4x4 matrix whose upper 3x4 elements are equal to the 3x4 transformation matrix argument and whose bottom row is equal to (0,0,0,1).

vmathSoaM4MakeFrustum

Construct a perspective projection matrix based on frustum.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeFrustum(
        VmathSoaMatrix4 *result,
        vec_float4 left,
        vec_float4 right,
        vec_float4 bottom,
        vec_float4 top,
        vec_float4 zNear,
        vec_float4 zFar
);
```

Arguments

result	The constructed 4x4 matrix
left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

None

Description

Construct a perspective projection matrix based on frustum, equal to:

```
2*zNear/(right-left)   0          (right+left)/(right-left)      0
       0       2*zNear/(top-bottom)  (top+bottom)/(top-bottom)      0
       0               0         -(zFar+zNear)/(zFar-zNear)
-2*zFar*zNear/(zFar-zNear)
       0               0                  -1                0  .
```

vmathSoaM4MakeIdentity

Construct an identity 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeIdentity(
        VmathSoaMatrix4 *result
);
```

Arguments

result The constructed 4x4 matrix

Return Values

None

Description

Construct an identity 4x4 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathSoaM4MakeLookAt

Construct viewing matrix based on eye position, position looked at, and up direction.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeLookAt(
        VmathSoaMatrix4 *result,
        const VmathSoaPoint3 *eyePos,
        const VmathSoaPoint3 *lookAtPos,
        const VmathSoaVector3 *upVec
);
```

Arguments

result	The constructed 4x4 matrix
eyePos	3-D point
lookAtPos	3-D point
upVec	3-D vector

Return Values

None

Description

Construct the inverse of a coordinate frame that is centered at the eye position, with z axis directed away from lookAtPos, and y axis oriented to best match the up direction.

vmathSoaM4MakeOrthographic

Construct an orthographic projection matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeOrthographic(
        VmathSoaMatrix4 *result,
        vec_float4 left,
        vec_float4 right,
        vec_float4 bottom,
        vec_float4 top,
        vec_float4 zNear,
        vec_float4 zFar
);
```

Arguments

result	The constructed 4x4 matrix
left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

None

Description

Construct an orthographic projection matrix, equal to

```
2/(right-left)        0              0          -(right+left)/(right-left)
     0         2/(top-bottom)        0          -(top+bottom)/(top-bottom)
     0                0       -2/(zFar-zNear)   -(zFar+zNear)/(zFar-zNear)
     0                0              0                      1   .
```

vmathSoaM4MakePerspective

Construct a perspective projection matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakePerspective(
        VmathSoaMatrix4 *result,
        vec_float4 fovyRadians,
        vec_float4 aspect,
        vec_float4 zNear,
        vec_float4 zFar
);
```

Arguments

result	The constructed 4x4 matrix
fovyRadians	Scalar value
aspect	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

None

Description

Construct a perspective projection matrix, equal to:

```
cot(fovyRadians/2)/aspect   0                 0                    0
       0             cot(fovyRadians/2)       0                    0
       0                    0    (zFar+zNear)/(zNear-zFar)
2*zFar*zNear/(zNear-zFar)
       0                    0            -1                   0  .
```

vmathSoaM4MakeRotationAxis

Construct a 4x4 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeRotationAxis(
        VmathSoaMatrix4 *result,
        vec_float4 radians,
        const VmathSoaVector3 *unitVec
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a 4x4 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaM4MakeRotationQ

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeRotationQ(
        VmathSoaMatrix4 *result,
        const VmathSoaQuat *unitQuat
);
```

Arguments

result	A 4x4 matrix that applies the same rotation as *unitQuat*
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 4x4 matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaM4MakeRotationX

Construct a 4x4 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeRotationX(
        VmathSoaMatrix4 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value

Return Values

None

Description

Construct a 4x4 matrix to rotate around the x axis by the specified radians angle.

vmathSoaM4MakeRotationY

Construct a 4x4 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeRotationY(
        VmathSoaMatrix4 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value

Return Values

None

Description

Construct a 4x4 matrix to rotate around the y axis by the specified radians angle.

vmathSoaM4MakeRotationZ

Construct a 4x4 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeRotationZ(
        VmathSoaMatrix4 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 4x4 matrix
radians	Scalar value

Return Values

None

Description

Construct a 4x4 matrix to rotate around the z axis by the specified radians angle.

vmathSoaM4MakeRotationZYX

Construct a 4x4 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeRotationZYX(
        VmathSoaMatrix4 *result,
        const VmathSoaVector3 *radiansXYZ
);
```

Arguments

result	The constructed 4x4 matrix
radiansXYZ	3-D vector

Return Values

None

Description

Construct a 4x4 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to *rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())*.

vmathSoaM4MakeScale

Construct a 4x4 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeScale(
        VmathSoaMatrix4 *result,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	The constructed 4x4 matrix
scaleVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathSoaM4MakeTranslation

Construct a 4x4 matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MakeTranslation(
        VmathSoaMatrix4 *result,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	The constructed 4x4 matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 4x4 matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathSoaM4Mul

Multiply two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Mul(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat0,
        const VmathSoaMatrix4 *mat1
);
```

Arguments

result	Product of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Multiply two 4x4 matrices.

vmathSoaM4MulP3

Multiply a 4x4 matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MulP3(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	Product of the specified 4x4 matrix and 3-D point
mat	4x4 matrix
pnt	3-D point

Return Values

None

Description

Multiply a 4x4 matrix by a 3-D point treated as if it were a 4-D vector with the w element equal to 1.

vmathSoaM4MulPerElem

Multiply two 4x4 matrices per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MulPerElem(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat0,
        const VmathSoaMatrix4 *mat1
);
```

Arguments

result	4x4 matrix in which each element is the product of the corresponding elements of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Multiply two 4x4 matrices element by element.

vmathSoaM4MulT3

Multiply a 4x4 matrix by a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MulT3(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Product of the specified 4x4 matrix and 3x4 transformation matrix
mat	4x4 matrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Multiply a 4x4 matrix by a 3x4 transformation matrix treated as if it were a 4x4 matrix with the bottom row equal to (0,0,0,1).

vmathSoaM4MulV3

Multiply a 4x4 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MulV3(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat,
        const VmathSoaVector3 *vec
);
```

Arguments

result	Product of the specified 4x4 matrix and 3-D vector
mat	4x4 matrix
vec	3-D vector

Return Values

None

Description

Multiply a 4x4 matrix by a 3-D vector treated as if it were a 4-D vector with the w element equal to 0.

vmathSoaM4MulV4

Multiply a 4x4 matrix by a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4MulV4(
        VmathSoaVector4 *result,
        const VmathSoaMatrix4 *mat,
        const VmathSoaVector4 *vec
);
```

Arguments

result	Product of the specified 4x4 matrix and 4-D vector
mat	4x4 matrix
vec	4-D vector

Return Values

None

Description

Multiply a 4x4 matrix by a 4-D vector.

vmathSoaM4Neg

Negate all elements of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Neg(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	4x4 matrix containing negated elements of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Negate all elements of a 4x4 matrix.

vmathSoaM4OrthoInverse

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix with an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4OrthoInverse(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	Inverse of the specified 4x4 matrix
mat	4x4 matrix

Return Values

None

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), whose translation vector is -transpose(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions.

vmathSoaM4PrependScale

Prepend (pre-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4PrependScale(
        VmathSoaMatrix4 *result,
        const VmathSoaVector3 *scaleVec,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	The product of a scale transformation created from `scaleVec` and `mat`
scaleVec	3-D vector
mat	4x4 matrix

Return Values

None

Description

Pre-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM4Print

Print a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Print(
        const VmathSoaMatrix4 *mat
);
```

Arguments

mat 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM4Prints

Print a 4x4 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Prints(
        const VmathSoaMatrix4 *mat,
        const char *name
);
```

Arguments

mat	4x4 matrix
name	String printed with the 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix and an associated string identifier. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM4ScalarMul

Multiply a 4x4 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4ScalarMul(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat,
        vec_float4 scalar
);
```

Arguments

result	Product of the specified 4x4 matrix and scalar
mat	4x4 matrix
scalar	Scalar value

Return Values

None

Description

Multiply a 4x4 matrix by a scalar.

vmathSoaM4Select

Conditionally select between two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Select(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat0,
        const VmathSoaMatrix4 *mat1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the 4x4 matrix at the corresponding slot of *mat0* or *mat1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *mat0* and a value of 0xFFFFFFFF selects the slot of *mat1*
mat0	4x4 matrix
mat1	4x4 matrix
select1	For each of the four word slots, this mask selects either the 4x4 matrix in the corresponding slot of *mat0* or the 4x4 matrix in the corresponding slot of *mat1*. A 0 bit selects from *mat0* whereas a 1 bit selects from *mat1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the 4x4 matrices at each of the corresponding slots of *mat0* or *mat1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaM4SetCol

Set the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetCol(
        VmathSoaMatrix4 *result,
        int col,
        const VmathSoaVector4 *vec
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the column of a 4x4 matrix referred to by the specified index.

vmathSoaM4SetCol0

Set column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetCol0(
        VmathSoaMatrix4 *result,
        const VmathSoaVector4 *col0
);
```

Arguments

result	An output 4x4 matrix
col0	4-D vector

Return Values

None

Description

Set column 0 of a 4x4 matrix.

vmathSoaM4SetCol1

Set column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetCol1(
        VmathSoaMatrix4 *result,
        const VmathSoaVector4 *col1
);
```

Arguments

result	An output 4x4 matrix
col1	4-D vector

Return Values

None

Description

Set column 1 of a 4x4 matrix.

vmathSoaM4SetCol2

Set column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetCol2(
        VmathSoaMatrix4 *result,
        const VmathSoaVector4 *col2
);
```

Arguments

result	An output 4x4 matrix
col2	4-D vector

Return Values

None

Description

Set column 2 of a 4x4 matrix.

vmathSoaM4SetCol3

Set column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetCol3(
        VmathSoaMatrix4 *result,
        const VmathSoaVector4 *col3
);
```

Arguments

result	An output 4x4 matrix
col3	4-D vector

Return Values

None

Description

Set column 3 of a 4x4 matrix.

vmathSoaM4SetElem

Set the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetElem(
        VmathSoaMatrix4 *result,
        int col,
        int row,
        vec_float4 val
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3
val	Scalar value

Return Values

None

Description

Set the element of a 4x4 matrix referred to by column and row indices.

vmathSoaM4SetRow

Set the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetRow(
        VmathSoaMatrix4 *result,
        int row,
        const VmathSoaVector4 *vec
);
```

Arguments

result	An output 4x4 matrix
row	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the row of a 4x4 matrix referred to by the specified index.

vmathSoaM4SetTranslation

Set translation component.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetTranslation(
        VmathSoaMatrix4 *result,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	An output 4x4 matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 4x4 matrix equal to the specified 3-D vector.

Notes

This function does not change the bottom row elements.

vmathSoaM4SetUpper3x3

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4SetUpper3x3(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix3 *mat3
);
```

Arguments

result	An output 4x4 matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 4x4 matrix equal to the specified 3x3 matrix.

Notes

This function does not change the bottom row elements.

vmathSoaM4Sub

Subtract a 4x4 matrix from another 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Sub(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat0,
        const VmathSoaMatrix4 *mat1
);
```

Arguments

result	Difference of the specified 4x4 matrices
mat0	4x4 matrix
mat1	4x4 matrix

Return Values

None

Description

Subtract a 4x4 matrix from another 4x4 matrix.

vmathSoaM4Transpose

Transpose of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaM4Transpose(
        VmathSoaMatrix4 *result,
        const VmathSoaMatrix4 *mat
);
```

Arguments

result	*mat* transposed
mat	4x4 matrix

Return Values

None

Description

Compute the transpose of a 4x4 matrix.

Transformation Functions (SoA, by reference)

vmathSoaT3AbsPerElem

Compute the absolute value of a 3x4 transformation matrix per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3AbsPerElem(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	3x4 transformation matrix in which each element is the absolute value of the corresponding element of the specified 3x4 transformation matrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Compute the absolute value of each element of a 3x4 transformation matrix.

vmathSoaT3AppendScale

Append (post-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3AppendScale(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	The product of `tfrm` and a scale transformation created from `scaleVec`
tfrm	3x4 transformation matrix
scaleVec	3-D vector

Return Values

None

Description

Post-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaT3Copy

Copy a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Copy(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	The constructed result
tfrm	3x4 transformation matrix

Return Values

None

Description

Construct a copy of a 3x4 transformation matrix.

vmathSoaT3Get4Aos

Extract four AoS 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Get4Aos(
        const VmathSoaTransform3 *tfrm,
        VmathTransform3 *result0,
        VmathTransform3 *result1,
        VmathTransform3 *result2,
        VmathTransform3 *result3
);
```

Arguments

tfrm	3x4 transformation matrix
result0	An output AoS 3x4 transformation matrix
result1	An output AoS 3x4 transformation matrix
result2	An output AoS 3x4 transformation matrix
result3	An output AoS 3x4 transformation matrix

Return Values

None

Description

Extract four AoS 3x4 transformation matrices from four slots of an SoA 3x4 transformation matrix (transpose the data format).

vmathSoaT3GetCol

Get the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetCol(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm,
        int col
);
```

Arguments

result	The column referred to by the specified index
tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3

Return Values

None

Description

Get the column of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3GetCol0

Get column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetCol0(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Column 0
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 0 of a 3x4 transformation matrix.

vmathSoaT3GetCol1

Get column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetCol1(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Column 1
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 1 of a 3x4 transformation matrix.

vmathSoaT3GetCol2

Get column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetCol2(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Column 2
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 2 of a 3x4 transformation matrix.

vmathSoaT3GetCol3

Get column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetCol3(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Column 3
tfrm	3x4 transformation matrix

Return Values

None

Description

Get column 3 of a 3x4 transformation matrix.

vmathSoaT3GetElem

Get the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline vec_float4 vmathSoaT3GetElem(
        const VmathSoaTransform3 *tfrm,
        int col,
        int row
);
```

Arguments

tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x4 transformation matrix referred to by column and row indices.

vmathSoaT3GetRow

Get the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetRow(
        VmathSoaVector4 *result,
        const VmathSoaTransform3 *tfrm,
        int row
);
```

Arguments

result	The row referred to by the specified index
tfrm	3x4 transformation matrix
row	Index, expected in the range 0-2

Return Values

None

Description

Get the row of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3GetTranslation

Get the translation component of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetTranslation(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Translation component
tfrm	3x4 transformation matrix

Return Values

None

Description

Get the translation component of a 3x4 transformation matrix.

vmathSoaT3GetUpper3x3

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3GetUpper3x3(
        VmathSoaMatrix3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Upper-left 3x3 submatrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

vmathSoaT3Inverse

Inverse of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Inverse(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Inverse of *tfrm*
tfrm	3x4 transformation matrix

Return Values

None

Description

Compute the inverse of a 3x4 transformation matrix.

Notes

Result is unpredictable when the determinant of the left 3x3 submatrix is equal to or near 0.

vmathSoaT3MakeFrom4Aos

Insert four AoS 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeFrom4Aos(
        VmathSoaTransform3 *result,
        const VmathTransform3 *tfrm0,
        const VmathTransform3 *tfrm1,
        const VmathTransform3 *tfrm2,
        const VmathTransform3 *tfrm3
);
```

Arguments

result	The constructed 3x4 transformation matrix
tfrm0	AoS 3x4 transformation matrix
tfrm1	AoS 3x4 transformation matrix
tfrm2	AoS 3x4 transformation matrix
tfrm3	AoS 3x4 transformation matrix

Return Values

None

Description

Insert four AoS 3x4 transformation matrices into four slots of an SoA 3x4 transformation matrix (transpose the data format).

vmathSoaT3MakeFromAos

Replicate an AoS 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeFromAos(
        VmathSoaTransform3 *result,
        const VmathTransform3 *tfrm
);
```

Arguments

result	The constructed 3x4 transformation matrix
tfrm	AoS 3x4 transformation matrix

Return Values

None

Description

Replicate an AoS 3x4 transformation matrix in all four slots of an SoA 3x4 transformation matrix.

vmathSoaT3MakeFromCols

Construct a 3x4 transformation matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeFromCols(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *col0,
        const VmathSoaVector3 *col1,
        const VmathSoaVector3 *col2,
        const VmathSoaVector3 *col3
);
```

Arguments

result	The 3x4 transformation matrix that contains the specified columns
col0	3-D vector
col1	3-D vector
col2	3-D vector
col3	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix containing the specified columns.

vmathSoaT3MakeFromM3V3

Construct a 3x4 transformation matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeFromM3V3(
        VmathSoaTransform3 *result,
        const VmathSoaMatrix3 *tfrm,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
tfrm	3x3 matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix whose upper 3x3 elements are equal to the 3x3 matrix argument and whose translation component is equal to the 3-D vector argument.

vmathSoaT3MakeFromQV3

Construct a 3x4 transformation matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeFromQV3(
        VmathSoaTransform3 *result,
        const VmathSoaQuat *unitQuat,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument and whose translation component is equal to the 3-D vector argument.

vmathSoaT3MakeFromScalar

Set all elements of a 3x4 transformation matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeFromScalar(
        VmathSoaTransform3 *result,
        vec_float4 scalar
);
```

Arguments

result	The constructed 3x4 transformation matrix
scalar	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix with all elements set to the scalar value argument.

vmathSoaT3MakeIdentity

Construct an identity 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeIdentity(
        VmathSoaTransform3 *result
);
```

Arguments

result The constructed 3x4 transformation matrix

Return Values

None

Description

Construct an identity 3x4 transformation matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathSoaT3MakeRotationAxis

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeRotationAxis(
        VmathSoaTransform3 *result,
        vec_float4 radians,
        const VmathSoaVector3 *unitVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaT3MakeRotationQ

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeRotationQ(
        VmathSoaTransform3 *result,
        const VmathSoaQuat *unitQuat
);
```

Arguments

result	A 3x4 transformation matrix that applies the same rotation as `unitQuat`
unitQuat	Quaternion, expected to be unit-length

Return Values

None

Description

Construct a 3x4 transformation matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaT3MakeRotationX

Construct a 3x4 transformation matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeRotationX(
        VmathSoaTransform3 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the x axis by the specified radians angle.

vmathSoaT3MakeRotationY

Construct a 3x4 transformation matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeRotationY(
        VmathSoaTransform3 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the y axis by the specified radians angle.

vmathSoaT3MakeRotationZ

Construct a 3x4 transformation matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeRotationZ(
        VmathSoaTransform3 *result,
        vec_float4 radians
);
```

Arguments

result	The constructed 3x4 transformation matrix
radians	Scalar value

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the z axis by the specified radians angle.

vmathSoaT3MakeRotationZYX

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeRotationZYX(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *radiansXYZ
);
```

Arguments

result	The constructed 3x4 transformation matrix
radiansXYZ	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to *rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())*.

vmathSoaT3MakeScale

Construct a 3x4 transformation matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeScale(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *scaleVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
scaleVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathSoaT3MakeTranslation

Construct a 3x4 transformation matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MakeTranslation(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	The constructed 3x4 transformation matrix
translateVec	3-D vector

Return Values

None

Description

Construct a 3x4 transformation matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathSoaT3Mul

Multiply two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Mul(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm0,
        const VmathSoaTransform3 *tfrm1
);
```

Arguments

result	Product of the specified 3x4 transformation matrices
tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix

Return Values

None

Description

Multiply two 3x4 transformation matrices.

vmathSoaT3MulP3

Multiply a 3x4 transformation matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MulP3(
        VmathSoaPoint3 *result,
        const VmathSoaTransform3 *tfrm,
        const VmathSoaPoint3 *pnt
);
```

Arguments

result	Product of the specified 3x4 transformation matrix and 3-D point
tfrm	3x4 transformation matrix
pnt	3-D point

Return Values

None

Description

Applies the 3x3 upper-left submatrix and the translation component of a 3x4 transformation matrix to a 3-D point.

vmathSoaT3MulPerElem

Multiply two 3x4 transformation matrices per element.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MulPerElem(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm0,
        const VmathSoaTransform3 *tfrm1
);
```

Arguments

result	3x4 transformation matrix in which each element is the product of the corresponding elements of the specified 3x4 transformation matrices
tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix

Return Values

None

Description

Multiply two 3x4 transformation matrices element by element.

vmathSoaT3MulV3

Multiply a 3x4 transformation matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3MulV3(
        VmathSoaVector3 *result,
        const VmathSoaTransform3 *tfrm,
        const VmathSoaVector3 *vec
);
```

Arguments

result	Product of the specified 3x4 transformation matrix and 3-D vector
tfrm	3x4 transformation matrix
vec	3-D vector

Return Values

None

Description

Applies the 3x3 upper-left submatrix (but not the translation component) of a 3x4 transformation matrix to a 3-D vector.

vmathSoaT3OrthoInverse

Compute the inverse of a 3x4 transformation matrix, expected to have an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3OrthoInverse(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	Inverse of the specified 3x4 transformation matrix
tfrm	3x4 transformation matrix

Return Values

None

Description

Naming the upper-left 3x3 submatrix of the specified 3x4 transformation matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), and whose translation vector is -transpose(M)*v.

Notes

This can be used to achieve better performance than a general inverse when the specified 3x4 transformation matrix meets the given restrictions.

vmathSoaT3PrependScale

Prepend (pre-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3PrependScale(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *scaleVec,
        const VmathSoaTransform3 *tfrm
);
```

Arguments

result	The product of a scale transformation created from *scaleVec* and *tfrm*
scaleVec	3-D vector
tfrm	3x4 transformation matrix

Return Values

None

Description

Pre-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaT3Print

Print a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Print(
        const VmathSoaTransform3 *tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaT3Prints

Print a 3x4 transformation matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Prints(
        const VmathSoaTransform3 *tfrm,
        const char *name
);
```

Arguments

tfrm	3x4 transformation matrix
name	String printed with the 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix and an associated string identifier. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaT3Select

Conditionally select between two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3Select(
        VmathSoaTransform3 *result,
        const VmathSoaTransform3 *tfrm0,
        const VmathSoaTransform3 *tfrm1,
        vec_uint4 select1
);
```

Arguments

result	Each slot of the result is equal to the 3x4 transformation matrix at the corresponding slot of *tfrm0* or *tfrm1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *tfrm0* and a value of 0xFFFFFFFF selects the slot of *tfrm1*
tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix
select1	For each of the four word slots, this mask selects either the 3x4 transformation matrix in the corresponding slot of *tfrm0* or the 3x4 transformation matrix in the corresponding slot of *tfrm1*. A 0 bit selects from *tfrm0* whereas a 1 bit selects from *tfrm1*. Identical bits should be set for each word of the mask.

Return Values

None

Description

Conditionally select one of the 3x4 transformation matrices at each of the corresponding slots of *tfrm0* or *tfrm1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaT3SetCol

Set the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetCol(
        VmathSoaTransform3 *result,
        int col,
        const VmathSoaVector3 *vec
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
vec	3-D vector

Return Values

None

Description

Set the column of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3SetCol0

Set column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetCol0(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *col0
);
```

Arguments

result	An output 3x4 transformation matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x4 transformation matrix.

vmathSoaT3SetCol1

Set column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetCol1(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *col1
);
```

Arguments

result	An output 3x4 transformation matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x4 transformation matrix.

vmathSoaT3SetCol2

Set column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetCol2(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *col2
);
```

Arguments

result	An output 3x4 transformation matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x4 transformation matrix.

vmathSoaT3SetCol3

Set column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetCol3(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *col3
);
```

Arguments

result	An output 3x4 transformation matrix
col3	3-D vector

Return Values

None

Description

Set column 3 of a 3x4 transformation matrix.

vmathSoaT3SetElem

Set the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetElem(
        VmathSoaTransform3 *result,
        int col,
        int row,
        vec_float4 val
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x4 transformation matrix referred to by column and row indices.

vmathSoaT3SetRow

Set the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetRow(
        VmathSoaTransform3 *result,
        int row,
        const VmathSoaVector4 *vec
);
```

Arguments

result	An output 3x4 transformation matrix
row	Index, expected in the range 0-2
vec	4-D vector

Return Values

None

Description

Set the row of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3SetTranslation

Set translation component.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetTranslation(
        VmathSoaTransform3 *result,
        const VmathSoaVector3 *translateVec
);
```

Arguments

result	An output 3x4 transformation matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 3x4 transformation matrix equal to the specified 3-D vector.

vmathSoaT3SetUpper3x3

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa.h>
static inline void vmathSoaT3SetUpper3x3(
        VmathSoaTransform3 *result,
        const VmathSoaMatrix3 *mat3
);
```

Arguments

result	An output 3x4 transformation matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 3x4 transformation matrix equal to the specified 3x3 matrix.

3-D Vector Functions (AoS, by value)

vmathV3AbsPerElem_V

Compute the absolute value of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3AbsPerElem_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the absolute value of the corresponding element of vec

Description

Compute the absolute value of each element of a 3-D vector.

vmathV3Add_V

Add two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Add_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Sum of the specified 3-D vectors

Description

Add two 3-D vectors.

vmathV3AddP3_V

Add a 3-D vector to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathV3AddP3_V(
        VmathVector3 vec,
        VmathPoint3 pnt
);
```

Arguments

vec 3-D vector
pnt 3-D point

Return Values

Sum of the specified 3-D vector and 3-D point

Description

Add a 3-D vector to a 3-D point.

vmathV3CopySignPerElem_V

Copy sign from one 3-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3CopySignPerElem_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathV3Cross_V

Compute cross product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Cross_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Cross product of the specified 3-D vectors

Description

Compute cross product of two 3-D vectors.

vmathV3CrossMatrix_V

Cross-product matrix of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathV3CrossMatrix_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Cross-product matrix of *vec*

Description

Compute a matrix that, when multiplied by a 3-D vector, produces the same result as a cross product with that 3-D vector.

vmathV3CrossMatrixMul_V

Create cross-product matrix and multiply.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathV3CrossMatrixMul_V(
        VmathVector3 vec,
        VmathMatrix3 mat
);
```

Arguments

vec 3-D vector
mat 3x3 matrix

Return Values

Product of cross-product matrix of *vec* and *mat*

Description

Multiply a cross-product matrix by another matrix.

Notes

Faster than separately creating a cross-product matrix and multiplying.

vmathV3DivPerElem_V

Divide two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3DivPerElem_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0 3-D vector
vec1 3-D vector

Return Values

3-D vector in which each element is the quotient of the corresponding elements of the specified 3-D vectors

Description

Divide two 3-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathV3Dot_V

Compute the dot product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3Dot_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0 3-D vector
vec1 3-D vector

Return Values

Dot product of the specified 3-D vectors

Description

Compute the dot product of two 3-D vectors.

vmathV3Get128_V

Get vector float data from a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline vec_float4 vmathV3Get128_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Internal vector float data

Description

Get internal vector float data from a 3-D vector.

vmathV3GetElem_V

Get an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3GetElem_V(
        VmathVector3 vec,
        int idx
);
```

Arguments

vec	3-D vector
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathV3GetX_V

Get the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3GetX_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

x element of a 3-D vector

Description

Get the x element of a 3-D vector.

vmathV3GetY_V

Get the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3GetY_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

y element of a 3-D vector

Description

Get the y element of a 3-D vector.

vmathV3GetZ_V

Get the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3GetZ_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

z element of a 3-D vector

Description

Get the z element of a 3-D vector.

vmathV3Length_V

Compute the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3Length_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Length of the specified 3-D vector

Description

Compute the length of a 3-D vector.

vmathV3LengthSqr_V

Compute the square of the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3LengthSqr_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Square of the length of the specified 3-D vector

Description

Compute the square of the length of a 3-D vector.

vmathV3Lerp_V

Linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Lerp_V(
        float t,
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

t	Interpolation parameter
vec0	3-D vector
vec1	3-D vector

Return Values

Interpolated 3-D vector

Description

Linearly interpolate between two 3-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathV3LoadXYZArray_V

Load four three-float 3-D vectors, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3LoadXYZArray_V(
        VmathVector3 *vec0,
        VmathVector3 *vec1,
        VmathVector3 *vec2,
        VmathVector3 *vec3,
        const vec_float4 *threeQuads
);
```

Arguments

vec0	An output 3-D vector
vec1	An output 3-D vector
vec2	An output 3-D vector
vec3	An output 3-D vector
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D vectors, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four 3-D vectors.

vmathV3MakeFrom128_V

Set vector float data in a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeFrom128_V(
        vec_float4 vf4
);
```

Arguments

vf4 Scalar value

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector whose internal vector float data is set to the vector float argument.

vmathV3MakeFromElems_V

Construct a 3-D vector from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeFromElems_V(
        float x,
        float y,
        float z
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value

Return Values

The 3-D vector that contains the specified elements

Description

Construct a 3-D vector containing the specified x, y, and z elements.

vmathV3MakeFromP3_V

Copy elements from a 3-D point into a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeFromP3_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector containing the x, y, and z elements of the specified 3-D point.

vmathV3MakeFromScalar_V

Set all elements of a 3-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector with all elements set to the scalar value argument.

vmathV3MakeXAxis_V

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeXAxis_V();
```

Arguments

None

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector equal to (1,0,0).

vmathV3MakeYAxis_V

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeYAxis_V();
```

Arguments

None

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector equal to (0,1,0).

vmathV3MakeZAxis_V

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MakeZAxis_V();
```

Arguments

None

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector equal to (0,0,1).

vmathV3MaxElem_V

Maximum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3MaxElem_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 3-D vector.

vmathV3MaxPerElem_V

Maximum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MaxPerElem_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors

Description

Create a 3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors.

vmathV3MinElem_V

Minimum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3MinElem_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 3-D vector.

vmathV3MinPerElem_V

Minimum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MinPerElem_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the minimum of the corresponding elements of the specified 3-D vectors

Description

Create a 3-D vector in which each element is the minimum of the corresponding elements of two specified 3-D vectors.

vmathV3MulPerElem_V

Multiply two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3MulPerElem_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the product of the corresponding elements of the specified 3-D vectors

Description

Multiply two 3-D vectors element by element.

vmathV3Neg_V

Negate all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Neg_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector containing negated elements of the specified 3-D vector

Description

Negate all elements of a 3-D vector.

vmathV3Normalize_V

Normalize a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Normalize_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

The specified 3-D vector scaled to unit length

Description

Compute a normalized 3-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathV3Outer_V

Outer product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathV3Outer_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

The 3x3 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*

Description

Compute the outer product of two 3-D vectors.

vmathV3Print_V

Print a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3Print_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

None

Description

Print a 3-D vector. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV3Prints_V

Print a 3-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3Prints_V(
        VmathVector3 vec,
        const char *name
);
```

Arguments

vec	3-D vector
name	String printed with the 3-D vector

Return Values

None

Description

Print a 3-D vector and an associated string identifier. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV3RecipPerElem_V

Compute the reciprocal of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3RecipPerElem_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector

Description

Create a 3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathV3RowMul_V

Pre-multiply a row vector by a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3RowMul_V(
        VmathVector3 vec,
        VmathMatrix3 mat
);
```

Arguments

vec	3-D vector
mat	3x3 matrix

Return Values

Product of a row-vector and a 3x3 matrix

Description

Transpose a 3-D vector into a row vector and pre-multiply by 3x3 matrix.

Notes

Slower than column post-multiply.

vmathV3RsqrtPerElem_V

Compute the reciprocal square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3RsqrtPerElem_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector

Description

Create a 3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathV3ScalarDiv_V

Divide a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3ScalarDiv_V(
        VmathVector3 vec,
        float scalar
);
```

Arguments

vec	3-D vector
scalar	Scalar value

Return Values

Quotient of the specified 3-D vector and scalar

Description

Divide a 3-D vector by a scalar.

vmathV3ScalarMul_V

Multiply a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3ScalarMul_V(
        VmathVector3 vec,
        float scalar
);
```

Arguments

vec	3-D vector
scalar	Scalar value

Return Values

Product of the specified 3-D vector and scalar

Description

Multiply a 3-D vector by a scalar.

vmathV3Select_V

Conditionally select between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Select_V(
        VmathVector3 vec0,
        VmathVector3 vec1,
        unsigned int select1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
select1	False selects the vec0 argument, true selects the vec1 argument

Return Values

Equal to *vec0* if *select1* == 0, or to *vec1* if *select1* != 0

Description

Conditionally select one of the 3-D vector arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathV3SetElem_V

Set an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3SetElem_V(
        VmathVector3 *result,
        int idx,
        float value
);
```

Arguments

result	An output 3-D vector
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathV3SetX_V

Set the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3SetX_V(
        VmathVector3 *result,
        float x
);
```

Arguments

result	An output 3-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D vector to the specified scalar value.

vmathV3SetY_V

Set the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3SetY_V(
        VmathVector3 *result,
        float y
);
```

Arguments

result	An output 3-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D vector to the specified scalar value.

vmathV3SetZ_V

Set the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3SetZ_V(
        VmathVector3 *result,
        float z
);
```

Arguments

result	An output 3-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D vector to the specified scalar value.

vmathV3Slerp_V

Spherical linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Slerp_V(
        float t,
        VmathVector3 unitVec0,
        VmathVector3 unitVec1
);
```

Arguments

t	Interpolation parameter
unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

Interpolated 3-D vector

Description

Perform spherical linear interpolation between two 3-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp *t* between 0 and 1.

vmathV3SqrtPerElem_V

Compute the square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3SqrtPerElem_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector

Description

Create a 3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathV3StoreHalfFloats_V

Store eight 3-D vectors as half-floats.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3StoreHalfFloats_V(
        VmathVector3 vec0,
        VmathVector3 vec1,
        VmathVector3 vec2,
        VmathVector3 vec3,
        VmathVector3 vec4,
        VmathVector3 vec5,
        VmathVector3 vec6,
        VmathVector3 vec7,
        vec_ushort8 *threeQuads
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
vec2	3-D vector
vec3	3-D vector
vec4	3-D vector
vec5	3-D vector
vec6	3-D vector
vec7	3-D vector
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight 3-D vectors in three quadwords of half-float values. The output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathV3StoreXYZ_V

Store x, y, and z elements of a 3-D vector in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3StoreXYZ_V(
        VmathVector3 vec,
        vec_float4 *quad
);
```

Arguments

vec	3-D vector
quad	Pointer to a quadword in which x, y, and z will be stored

Return Values

None

Description

Store x, y, and z elements of a 3-D vector in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

vmathV3StoreXYZArray_V

Store four 3-D vectors in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV3StoreXYZArray_V(
        VmathVector3 vec0,
        VmathVector3 vec1,
        VmathVector3 vec2,
        VmathVector3 vec3,
        vec_float4 *threeQuads
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
vec2	3-D vector
vec3	3-D vector
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four 3-D vectors in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathV3Sub_V

Subtract a 3-D vector from another 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV3Sub_V(
        VmathVector3 vec0,
        VmathVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Difference of the specified 3-D vectors

Description

Subtract a 3-D vector from another 3-D vector.

vmathV3Sum_V

Compute the sum of all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV3Sum_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 3-D vector.

4-D Vector Functions (AoS, by value)

vmathV4AbsPerElem_V

Compute the absolute value of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4AbsPerElem_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the absolute value of the corresponding element of vec

Description

Compute the absolute value of each element of a 4-D vector.

vmathV4Add_V

Add two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Add_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Sum of the specified 4-D vectors

Description

Add two 4-D vectors.

vmathV4CopySignPerElem_V

Copy sign from one 4-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4CopySignPerElem_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathV4DivPerElem_V

Divide two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4DivPerElem_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the quotient of the corresponding elements of the specified 4-D vectors

Description

Divide two 4-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathV4Dot_V

Compute the dot product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4Dot_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Dot product of the specified 4-D vectors

Description

Compute the dot product of two 4-D vectors.

vmathV4Get128_V

Get vector float data from a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline vec_float4 vmathV4Get128_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Internal vector float data

Description

Get internal vector float data from a 4-D vector.

vmathV4GetElem_V

Get an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4GetElem_V(
        VmathVector4 vec,
        int idx
);
```

Arguments

vec	4-D vector
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathV4GetW_V

Get the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4GetW_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

w element of a 4-D vector

Description

Get the w element of a 4-D vector.

vmathV4GetX_V

Get the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4GetX_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

x element of a 4-D vector

Description

Get the x element of a 4-D vector.

vmathV4GetXYZ_V

Get the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathV4GetXYZ_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

3-D vector containing x, y, and z elements

Description

Extract a 4-D vector's x, y, and z elements into a 3-D vector.

vmathV4GetY_V

Get the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4GetY_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

y element of a 4-D vector

Description

Get the y element of a 4-D vector.

vmathV4GetZ_V

Get the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4GetZ_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

z element of a 4-D vector

Description

Get the z element of a 4-D vector.

vmathV4Length_V

Compute the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4Length_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Length of the specified 4-D vector

Description

Compute the length of a 4-D vector.

vmathV4LengthSqr_V

Compute the square of the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4LengthSqr_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Square of the length of the specified 4-D vector

Description

Compute the square of the length of a 4-D vector.

vmathV4Lerp_V

Linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Lerp_V(
        float t,
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

t	Interpolation parameter
vec0	4-D vector
vec1	4-D vector

Return Values

Interpolated 4-D vector

Description

Linearly interpolate between two 4-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathV4MakeFrom128_V

Set vector float data in a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFrom128_V(
        vec_float4 vf4
);
```

Arguments

vf4 Scalar value

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector whose internal vector float data is set to the vector float argument.

vmathV4MakeFromElems_V

Construct a 4-D vector from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFromElems_V(
        float x,
        float y,
        float z,
        float w
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value
w Scalar value

Return Values

The 4-D vector that contains the specified elements

Description

Construct a 4-D vector containing the specified x, y, z, and w elements.

vmathV4MakeFromP3_V

Copy x, y, and z from a 3-D point into a 4-D vector, and set w to 1.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFromP3_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D point and with the w element set to 1.

vmathV4MakeFromQ_V

Copy elements from a quaternion into a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFromQ_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector containing the x, y, z, and w elements of the specified quaternion.

vmathV4MakeFromScalar_V

Set all elements of a 4-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with all elements set to the scalar value argument.

vmathV4MakeFromV3_V

Copy x, y, and z from a 3-D vector into a 4-D vector, and set w to 0.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFromV3_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to 0.

vmathV4MakeFromV3Scalar_V

Construct a 4-D vector from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeFromV3Scalar_V(
        VmathVector3 xyz,
        float w
);
```

Arguments

xyz	3-D vector
w	Scalar value

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathV4MakeWAxis_V

Construct w axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeWAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (0,0,0,1).

vmathV4MakeXAxis_V

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeXAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (1,0,0,0).

vmathV4MakeYAxis_V

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeYAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (0,1,0,0).

vmathV4MakeZAxis_V

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MakeZAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (0,0,1,0).

vmathV4MaxElem_V

Maximum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4MaxElem_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 4-D vector.

vmathV4MaxPerElem_V

Maximum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MaxPerElem_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors

Description

Create a 4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors.

vmathV4MinElem_V

Minimum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4MinElem_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 4-D vector.

vmathV4MinPerElem_V

Minimum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MinPerElem_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the minimum of the corresponding elements of the specified 4-D vectors

Description

Create a 4-D vector in which each element is the minimum of the corresponding elements of two specified 4-D vectors.

vmathV4MulPerElem_V

Multiply two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4MulPerElem_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the product of the corresponding elements of the specified 4-D vectors

Description

Multiply two 4-D vectors element by element.

vmathV4Neg_V

Negate all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Neg_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector containing negated elements of the specified 4-D vector

Description

Negate all elements of a 4-D vector.

vmathV4Normalize_V

Normalize a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Normalize_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

The specified 4-D vector scaled to unit length

Description

Compute a normalized 4-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathV4Outer_V

Outer product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathV4Outer_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

The 4x4 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*

Description

Compute the outer product of two 4-D vectors.

vmathV4Print_V

Print a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4Print_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

None

Description

Print a 4-D vector. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV4Prints_V

Print a 4-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4Prints_V(
        VmathVector4 vec,
        const char *name
);
```

Arguments

vec	4-D vector
name	String printed with the 4-D vector

Return Values

None

Description

Print a 4-D vector and an associated string identifier. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathV4RecipPerElem_V

Compute the reciprocal of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4RecipPerElem_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector

Description

Create a 4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathV4RsqrtPerElem_V

Compute the reciprocal square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4RsqrtPerElem_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector

Description

Create a 4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathV4ScalarDiv_V

Divide a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4ScalarDiv_V(
        VmathVector4 vec,
        float scalar
);
```

Arguments

vec	4-D vector
scalar	Scalar value

Return Values

Quotient of the specified 4-D vector and scalar

Description

Divide a 4-D vector by a scalar.

vmathV4ScalarMul_V

Multiply a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4ScalarMul_V(
        VmathVector4 vec,
        float scalar
);
```

Arguments

vec	4-D vector
scalar	Scalar value

Return Values

Product of the specified 4-D vector and scalar

Description

Multiply a 4-D vector by a scalar.

vmathV4Select_V

Conditionally select between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Select_V(
        VmathVector4 vec0,
        VmathVector4 vec1,
        unsigned int select1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector
select1	False selects the vec0 argument, true selects the vec1 argument

Return Values

Equal to *vec0* if *select1* == 0, or to *vec1* if *select1* != 0

Description

Conditionally select one of the 4-D vector arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathV4SetElem_V

Set an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4SetElem_V(
        VmathVector4 *result,
        int idx,
        float value
);
```

Arguments

result	An output 4-D vector
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathV4SetW_V

Set the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4SetW_V(
        VmathVector4 *result,
        float w
);
```

Arguments

result	An output 4-D vector
w	Scalar value

Return Values

None

Description

Set the w element of a 4-D vector to the specified scalar value.

vmathV4SetX_V

Set the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4SetX_V(
        VmathVector4 *result,
        float x
);
```

Arguments

result	An output 4-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 4-D vector to the specified scalar value.

vmathV4SetXYZ_V

Set the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4SetXYZ_V(
        VmathVector4 *result,
        VmathVector3 vec
);
```

Arguments

result	An output 4-D vector
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathV4SetY_V

Set the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4SetY_V(
        VmathVector4 *result,
        float y
);
```

Arguments

result	An output 4-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 4-D vector to the specified scalar value.

vmathV4SetZ_V

Set the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4SetZ_V(
        VmathVector4 *result,
        float z
);
```

Arguments

result	An output 4-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 4-D vector to the specified scalar value.

vmathV4Slerp_V

Spherical linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Slerp_V(
        float t,
        VmathVector4 unitVec0,
        VmathVector4 unitVec1
);
```

Arguments

t	Interpolation parameter
unitVec0	4-D vector, expected to be unit-length
unitVec1	4-D vector, expected to be unit-length

Return Values

Interpolated 4-D vector

Description

Perform spherical linear interpolation between two 4-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp `t` between 0 and 1.

vmathV4SqrtPerElem_V

Compute the square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4SqrtPerElem_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector

Description

Create a 4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathV4StoreHalfFloats_V

Store four 4-D vectors as half-floats.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathV4StoreHalfFloats_V(
        VmathVector4 vec0,
        VmathVector4 vec1,
        VmathVector4 vec2,
        VmathVector4 vec3,
        vec_ushort8 *twoQuads
);
```

Arguments

vec0	4-D vector
vec1	4-D vector
vec2	4-D vector
vec3	4-D vector
twoQuads	An output array of 2 quadwords containing 16 half-floats

Return Values

None

Description

Store four 4-D vectors in two quadwords of half-float values. The output is {x0,y0,z0,w0,x1,y1,z1,w1,x2,y2,z2,w2,x3,y3,z3,w3}.

vmathV4Sub_V

Subtract a 4-D vector from another 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathV4Sub_V(
        VmathVector4 vec0,
        VmathVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Difference of the specified 4-D vectors

Description

Subtract a 4-D vector from another 4-D vector.

vmathV4Sum_V

Compute the sum of all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathV4Sum_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 4-D vector.

Point Functions (AoS, by value)

vmathP3AbsPerElem_V

Compute the absolute value of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3AbsPerElem_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the absolute value of the corresponding element of pnt

Description

Compute the absolute value of each element of a 3-D point.

vmathP3AddV3_V

Add a 3-D point to a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3AddV3_V(
        VmathPoint3 pnt,
        VmathVector3 vec
);
```

Arguments

pnt	3-D point
vec	3-D vector

Return Values

Sum of the specified 3-D point and 3-D vector

Description

Add a 3-D point to a 3-D vector.

vmathP3CopySignPerElem_V

Copy sign from one 3-D point to another, per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3CopySignPerElem_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element has the magnitude of the corresponding element of *pnt0* and the sign of the corresponding element of *pnt1*

Description

For each element, create a value composed of the magnitude of *pnt0* and the sign of *pnt1*.

vmathP3Dist_V

Compute the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3Dist_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Distance between two 3-D points

Description

Compute the distance between two 3-D points.

vmathP3DistFromOrigin_V

Compute the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3DistFromOrigin_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Distance of a 3-D point from the origin

Description

Compute the distance of a 3-D point from the coordinate-system origin.

vmathP3DistSqr_V

Compute the square of the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3DistSqr_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Square of the distance between two 3-D points

Description

Compute the square of the distance between two 3-D points.

vmathP3DistSqrFromOrigin_V

Compute the square of the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3DistSqrFromOrigin_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Square of the distance of a 3-D point from the origin

Description

Compute the square of the distance of a 3-D point from the coordinate-system origin.

vmathP3DivPerElem_V

Divide two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3DivPerElem_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the quotient of the corresponding elements of the specified 3-D points

Description

Divide two 3-D points element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathP3Get128_V

Get vector float data from a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline vec_float4 vmathP3Get128_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Internal vector float data

Description

Get internal vector float data from a 3-D point.

vmathP3GetElem_V

Get an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3GetElem_V(
        VmathPoint3 pnt,
        int idx
);
```

Arguments

pnt 3-D point
idx Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathP3GetX_V

Get the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3GetX_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

x element of a 3-D point

Description

Get the x element of a 3-D point.

vmathP3GetY_V

Get the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3GetY_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

y element of a 3-D point

Description

Get the y element of a 3-D point.

vmathP3GetZ_V

Get the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3GetZ_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

z element of a 3-D point

Description

Get the z element of a 3-D point.

vmathP3Lerp_V

Linear interpolation between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3Lerp_V(
        float t,
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

t	Interpolation parameter
pnt0	3-D point
pnt1	3-D point

Return Values

Interpolated 3-D point

Description

Linearly interpolate between two 3-D points.

Notes

Does not clamp *t* between 0 and 1.

vmathP3LoadXYZArray_V

Load four three-float 3-D points, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3LoadXYZArray_V(
        VmathPoint3 *pnt0,
        VmathPoint3 *pnt1,
        VmathPoint3 *pnt2,
        VmathPoint3 *pnt3,
        const vec_float4 *threeQuads
);
```

Arguments

pnt0	An output 3-D point
pnt1	An output 3-D point
pnt2	An output 3-D point
pnt3	An output 3-D point
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D points, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four 3-D points.

vmathP3MakeFrom128_V

Set vector float data in a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MakeFrom128_V(
        vec_float4 vf4
);
```

Arguments

vf4 Scalar value

Return Values

The constructed 3-D point

Description

Construct a 3-D point whose internal vector float data is set to the vector float argument.

vmathP3MakeFromElems_V

Construct a 3-D point from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MakeFromElems_V(
        float x,
        float y,
        float z
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value

Return Values

The 3-D point that contains the specified elements

Description

Construct a 3-D point containing the specified x, y, and z elements.

vmathP3MakeFromScalar_V

Set all elements of a 3-D point to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3-D point

Description

Construct a 3-D point with all elements set to the scalar value argument.

vmathP3MakeFromV3_V

Copy elements from a 3-D vector into a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MakeFromV3_V(
        VmathVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

The constructed 3-D point

Description

Construct a 3-D point containing the x, y, and z elements of the specified 3-D vector.

vmathP3MaxElem_V

Maximum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3MaxElem_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Maximum value of all elements of *pnt*

Description

Compute the maximum value of all elements of a 3-D point.

vmathP3MaxPerElem_V

Maximum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MaxPerElem_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points

Description

Create a 3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points.

vmathP3MinElem_V

Minimum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3MinElem_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Minimum value of all elements of *pnt*

Description

Compute the minimum value of all elements of a 3-D point.

vmathP3MinPerElem_V

Minimum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MinPerElem_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the minimum of the corresponding elements of the specified 3-D points

Description

Create a 3-D point in which each element is the minimum of the corresponding elements of two specified 3-D points.

vmathP3MulPerElem_V

Multiply two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3MulPerElem_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the product of the corresponding elements of the specified 3-D points

Description

Multiply two 3-D points element by element.

vmathP3NonUniformScale_V

Apply non-uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3NonUniformScale_V(
        VmathPoint3 pnt,
        VmathVector3 scaleVec
);
```

Arguments

pnt	3-D point
scaleVec	3-D vector

Return Values

3-D point in which each element is the product of the corresponding elements of the specified 3-D point and 3-D vector

Description

Apply non-uniform scale to a 3-D point.

vmathP3Print_V

Print a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3Print_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

None

Description

Print a 3-D point. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathP3Prints_V

Print a 3-D point and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3Prints_V(
        VmathPoint3 pnt,
        const char *name
);
```

Arguments

pnt	3-D point
name	String printed with the 3-D point

Return Values

None

Description

Print a 3-D point and an associated string identifier. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathP3Projection_V

Scalar projection of a 3-D point on a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3Projection_V(
        VmathPoint3 pnt,
        VmathVector3 unitVec
);
```

Arguments

pnt	3-D point
unitVec	3-D vector, expected to be unit-length

Return Values

Scalar projection of the 3-D point on the unit-length 3-D vector

Description

Scalar projection of a 3-D point on a unit-length 3-D vector (dot product).

vmathP3RecipPerElem_V

Compute the reciprocal of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3RecipPerElem_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point

Description

Create a 3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function recipf4.

vmathP3RsqrtPerElem_V

Compute the reciprocal square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3RsqrtPerElem_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point

Description

Create a 3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathP3Scale_V

Apply uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3Scale_V(
        VmathPoint3 pnt,
        float scaleVal
);
```

Arguments

pnt	3-D point
scaleVal	Scalar value

Return Values

3-D point in which every element is multiplied by the scalar value

Description

Apply uniform scale to a 3-D point.

vmathP3Select_V

Conditionally select between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3Select_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1,
        unsigned int select1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
select1	False selects the pnt0 argument, true selects the pnt1 argument

Return Values

Equal to *pnt0* if *select1* == 0, or to *pnt1* if *select1* != 0

Description

Conditionally select one of the 3-D point arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathP3SetElem_V

Set an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3SetElem_V(
        VmathPoint3 *result,
        int idx,
        float value
);
```

Arguments

result	An output 3-D point
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathP3SetX_V

Set the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3SetX_V(
        VmathPoint3 *result,
        float x
);
```

Arguments

result	An output 3-D point
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D point to the specified scalar value.

vmathP3SetY_V

Set the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3SetY_V(
        VmathPoint3 *result,
        float y
);
```

Arguments

result	An output 3-D point
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D point to the specified scalar value.

vmathP3SetZ_V

Set the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3SetZ_V(
        VmathPoint3 *result,
        float z
);
```

Arguments

result	An output 3-D point
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D point to the specified scalar value.

vmathP3SqrtPerElem_V

Compute the square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3SqrtPerElem_V(
        VmathPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the square root of the corresponding element of the specified 3-D point

Description

Create a 3-D point in which each element is the square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathP3StoreHalfFloats_V

Store eight 3-D points as half-floats.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3StoreHalfFloats_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1,
        VmathPoint3 pnt2,
        VmathPoint3 pnt3,
        VmathPoint3 pnt4,
        VmathPoint3 pnt5,
        VmathPoint3 pnt6,
        VmathPoint3 pnt7,
        vec_ushort8 *threeQuads
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
pnt2	3-D point
pnt3	3-D point
pnt4	3-D point
pnt5	3-D point
pnt6	3-D point
pnt7	3-D point
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight 3-D points in three quadwords of half-float values. The output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathP3StoreXYZ_V

Store x, y, and z elements of a 3-D point in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3StoreXYZ_V(
        VmathPoint3 pnt,
        vec_float4 *quad
);
```

Arguments

pnt	3-D point
quad	Pointer to a quadword in which x, y, and z will be stored

Return Values

None

Description

Store x, y, and z elements of a 3-D point in the first three words of a quadword. The value of the fourth word (the word with the highest address) remains unchanged.

vmathP3StoreXYZArray_V

Store four 3-D points in three quadwords.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathP3StoreXYZArray_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1,
        VmathPoint3 pnt2,
        VmathPoint3 pnt3,
        vec_float4 *threeQuads
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
pnt2	3-D point
pnt3	3-D point
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four 3-D points in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathP3Sub_V

Subtract a 3-D point from another 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathP3Sub_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Difference of the specified 3-D points

Description

Subtract a 3-D point from another 3-D point.

vmathP3SubV3_V

Subtract a 3-D vector from a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathP3SubV3_V(
        VmathPoint3 pnt,
        VmathVector3 vec
);
```

Arguments

pnt 3-D point
vec 3-D vector

Return Values

Difference of the specified 3-D point and 3-D vector

Description

Subtract a 3-D vector from a 3-D point.

vmathP3Sum_V

Compute the sum of all elements of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathP3Sum_V(
        VmathPoint3 pnt
);
```

Arguments

`pnt` 3-D point

Return Values

Sum of all elements of `pnt`

Description

Compute the sum of all elements of a 3-D point.

Quaternion Functions (AoS, by value)

vmathQAdd_V

Add two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQAdd_V(
        VmathQuat quat0,
        VmathQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Sum of the specified quaternions

Description

Add two quaternions.

vmathQConj_V

Compute the conjugate of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQConj_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

Conjugate of the specified quaternion

Description

Compute the conjugate of a quaternion.

vmathQDot_V

Compute the dot product of two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQDot_V(
        VmathQuat quat0,
        VmathQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Dot product of the specified quaternions

Description

Compute the dot product of two quaternions.

vmathQGet128_V

Get vector float data from a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline vec_float4 vmathQGet128_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

Internal vector float data

Description

Get internal vector float data from a quaternion.

vmathQGetElem_V

Get an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQGetElem_V(
        VmathQuat quat,
        int idx
);
```

Arguments

quat	Quaternion
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathQGetW_V

Get the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQGetW_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

w element of a quaternion

Description

Get the w element of a quaternion.

vmathQGetX_V

Get the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQGetX_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

x element of a quaternion

Description

Get the x element of a quaternion.

vmathQGetXYZ_V

Get the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathQGetXYZ_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

3-D vector containing x, y, and z elements

Description

Extract a quaternion's x, y, and z elements into a 3-D vector.

vmathQGetY_V

Get the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQGetY_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

y element of a quaternion

Description

Get the y element of a quaternion.

vmathQGetZ_V

Get the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQGetZ_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

z element of a quaternion

Description

Get the z element of a quaternion.

vmathQLength_V

Compute the length of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQLength_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

Length of the specified quaternion

Description

Compute the length of a quaternion.

vmathQLerp_V

Linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQLerp_V(
        float t,
        VmathQuat quat0,
        VmathQuat quat1
);
```

Arguments

t	Interpolation parameter
quat0	Quaternion
quat1	Quaternion

Return Values

Interpolated quaternion

Description

Linearly interpolate between two quaternions.

Notes

Does not clamp *t* between 0 and 1.

vmathQMakeFrom128_V

Set vector float data in a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeFrom128_V(
        vec_float4 vf4
);
```

Arguments

vf4 Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion whose internal vector float data is set to the vector float argument.

vmathQMakeFromElems_V

Construct a quaternion from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeFromElems_V(
        float x,
        float y,
        float z,
        float w
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value
w Scalar value

Return Values

The quaternion that contains the specified elements

Description

Construct a quaternion containing the specified x, y, z, and w elements.

vmathQMakeFromM3_V

Convert a rotation matrix to a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeFromM3_V(
        VmathMatrix3 rotMat
);
```

Arguments

rotMat	3x3 matrix, expected to be a rotation matrix

Return Values

The constructed quaternion

Description

Construct a unit-length quaternion representing the same transformation as a rotation matrix.

vmathQMakeFromScalar_V

Set all elements of a quaternion to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion with all elements set to the scalar value argument.

vmathQMakeFromV3Scalar_V

Construct a quaternion from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeFromV3Scalar_V(
        VmathVector3 xyz,
        float w
);
```

Arguments

xyz	3-D vector
w	Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathQMakeFromV4_V

Copy elements from a 4-D vector into a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeFromV4_V(
        VmathVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

The constructed quaternion

Description

Construct a quaternion containing the x, y, z, and w elements of the specified 4-D vector.

vmathQMakeIdentity_V

Construct an identity quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeIdentity_V();
```

Arguments

None

Return Values

The constructed quaternion

Description

Construct an identity quaternion equal to (0,0,0,1).

vmathQMakeRotationArc_V

Construct a quaternion to rotate between two unit-length 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeRotationArc_V(
        VmathVector3 unitVec0,
        VmathVector3 unitVec1
);
```

Arguments

unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate between two unit-length 3-D vectors.

Notes

The result is unpredictable if *unitVec0* and *unitVec1* point in opposite directions.

vmathQMakeRotationAxis_V

Construct a quaternion to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeRotationAxis_V(
        float radians,
        VmathVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around a unit-length 3-D vector by the specified radians angle.

vmathQMakeRotationX_V

Construct a quaternion to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeRotationX_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around the x axis by the specified radians angle.

vmathQMakeRotationY_V

Construct a quaternion to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeRotationY_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around the y axis by the specified radians angle.

vmathQMakeRotationZ_V

Construct a quaternion to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMakeRotationZ_V(
        float radians
);
```

Arguments

radians	Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around the z axis by the specified radians angle.

vmathQMul_V

Multiply two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQMul_V(
        VmathQuat quat0,
        VmathQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Product of the specified quaternions

Description

Multiply two quaternions.

vmathQNeg_V

Negate all elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQNeg_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

Quaternion containing negated elements of the specified quaternion

Description

Negate all elements of a quaternion.

vmathQNorm_V

Compute the norm of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathQNorm_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

The norm of the specified quaternion

Description

Compute the norm, equal to the square of the length, of a quaternion.

vmathQNormalize_V

Normalize a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQNormalize_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

The specified quaternion scaled to unit length

Description

Compute a normalized quaternion.

Notes

The result is unpredictable when all elements of quat are at or near zero.

vmathQPrint_V

Print a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQPrint_V(
        VmathQuat quat
);
```

Arguments

quat Quaternion

Return Values

None

Description

Print a quaternion.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathQPrints_V

Print a quaternion and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQPrints_V(
        VmathQuat quat,
        const char *name
);
```

Arguments

quat	Quaternion
name	String printed with the quaternion

Return Values

None

Description

Print a quaternion and an associated string identifier.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathQRotate_V

Use a unit-length quaternion to rotate a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathQRotate_V(
        VmathQuat unitQuat,
        VmathVector3 vec
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length
vec	3-D vector

Return Values

The rotated 3-D vector, equivalent to unitQuat*Quat(vec,0)*conj(unitQuat)

Description

Rotate a 3-D vector by applying a unit-length quaternion.

vmathQScalarDiv_V

Divide a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQScalarDiv_V(
        VmathQuat quat,
        float scalar
);
```

Arguments

quat	Quaternion
scalar	Scalar value

Return Values

Quotient of the specified quaternion and scalar

Description

Divide a quaternion by a scalar.

vmathQScalarMul_V

Multiply a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQScalarMul_V(
        VmathQuat quat,
        float scalar
);
```

Arguments

quat	Quaternion
scalar	Scalar value

Return Values

Product of the specified quaternion and scalar

Description

Multiply a quaternion by a scalar.

vmathQSelect_V

Conditionally select between two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQSelect_V(
        VmathQuat quat0,
        VmathQuat quat1,
        unsigned int select1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion
select1	False selects the quat0 argument, true selects the quat1 argument

Return Values

Equal to *quat0* if *select1* == 0, or to *quat1* if *select1* != 0

Description

Conditionally select one of the quaternion arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of *select1* to a VMX register may use more processing time than a branch.

vmathQSetElem_V

Set an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQSetElem_V(
        VmathQuat *result,
        int idx,
        float value
);
```

Arguments

result	An output quaternion
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathQSetW_V

Set the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQSetW_V(
        VmathQuat *result,
        float w
);
```

Arguments

result	An output quaternion
w	Scalar value

Return Values

None

Description

Set the w element of a quaternion to the specified scalar value.

vmathQSetX_V

Set the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQSetX_V(
        VmathQuat *result,
        float x
);
```

Arguments

result	An output quaternion
x	Scalar value

Return Values

None

Description

Set the x element of a quaternion to the specified scalar value.

vmathQSetXYZ_V

Set the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQSetXYZ_V(
        VmathQuat *result,
        VmathVector3 vec
);
```

Arguments

result	An output quaternion
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathQSetY_V

Set the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQSetY_V(
        VmathQuat *result,
        float y
);
```

Arguments

result	An output quaternion
y	Scalar value

Return Values

None

Description

Set the y element of a quaternion to the specified scalar value.

vmathQSetZ_V

Set the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathQSetZ_V(
        VmathQuat *result,
        float z
);
```

Arguments

result	An output quaternion
z	Scalar value

Return Values

None

Description

Set the z element of a quaternion to the specified scalar value.

vmathQSlerp_V

Spherical linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQSlerp_V(
        float t,
        VmathQuat unitQuat0,
        VmathQuat unitQuat1
);
```

Arguments

t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length

Return Values

Interpolated quaternion

Description

Perform spherical linear interpolation between two quaternions.

Notes

Interpolates along the shortest path between orientations. Does not clamp `t` between 0 and 1.

vmathQSquad_V

Spherical quadrangle interpolation.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQSquad_V(
        float t,
        VmathQuat unitQuat0,
        VmathQuat unitQuat1,
        VmathQuat unitQuat2,
        VmathQuat unitQuat3
);
```

Arguments

t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length
unitQuat2	Quaternion, expected to be unit-length
unitQuat3	Quaternion, expected to be unit-length

Return Values

Interpolated quaternion

Description

Perform spherical quadrangle interpolation between four quaternions.

vmathQSub_V

Subtract a quaternion from another quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathQuat vmathQSub_V(
        VmathQuat quat0,
        VmathQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Difference of the specified quaternions

Description

Subtract a quaternion from another quaternion.

3x3 Matrix Functions (AoS, by value)

vmathM3AbsPerElem_V

Compute the absolute value of a 3x3 matrix per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3AbsPerElem_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

3x3 matrix in which each element is the absolute value of the corresponding element of the specified 3x3 matrix

Description

Compute the absolute value of each element of a 3x3 matrix.

vmathM3Add_V

Add two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Add_V(
        VmathMatrix3 mat0,
        VmathMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

Sum of the specified 3x3 matrices

Description

Add two 3x3 matrices.

vmathM3AppendScale_V

Append (post-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3AppendScale_V(
        VmathMatrix3 mat,
        VmathVector3 scaleVec
);
```

Arguments

mat	3x3 matrix
scaleVec	3-D vector

Return Values

The product of *mat* and a scale transformation created from *scaleVec*

Description

Post-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM3Determinant_V

Determinant of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathM3Determinant_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 3x3 matrix.

vmathM3GetCol0_V

Get column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM3GetCol0_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Column 0

Description

Get column 0 of a 3x3 matrix.

vmathM3GetCol1_V

Get column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM3GetCol1_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Column 1

Description

Get column 1 of a 3x3 matrix.

vmathM3GetCol2_V

Get column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM3GetCol2_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Column 2

Description

Get column 2 of a 3x3 matrix.

vmathM3GetCol_V

Get the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM3GetCol_V(
        VmathMatrix3 mat,
        int col
);
```

Arguments

mat	3x3 matrix
col	Index, expected in the range 0-2

Return Values

The column referred to by the specified index

Description

Get the column of a 3x3 matrix referred to by the specified index.

vmathM3GetElem_V

Get the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathM3GetElem_V(
        VmathMatrix3 mat,
        int col,
        int row
);
```

Arguments

mat	3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2

Return Values

Element selected by `col` and `row`

Description

Get the element of a 3x3 matrix referred to by column and row indices.

vmathM3GetRow_V

Get the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM3GetRow_V(
        VmathMatrix3 mat,
        int row
);
```

Arguments

mat	3x3 matrix
row	Index, expected in the range 0-2

Return Values

The row referred to by the specified index

Description

Get the row of a 3x3 matrix referred to by the specified index.

vmathM3Inverse_V

Compute the inverse of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Inverse_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Inverse of *mat*

Description

Compute the inverse of a 3x3 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathM3MakeFromCols_V

Construct a 3x3 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeFromCols_V(
        VmathVector3 col0,
        VmathVector3 col1,
        VmathVector3 col2
);
```

Arguments

col0	3-D vector
col1	3-D vector
col2	3-D vector

Return Values

The 3x3 matrix that contains the specified columns

Description

Construct a 3x3 matrix containing the specified columns.

vmathM3MakeFromQ_V

Construct a 3x3 rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeFromQ_V(
        VmathQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 3x3 matrix that applies the same rotation as *unitQuat*

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathM3MakeFromScalar_V

Set all elements of a 3x3 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix with all elements set to the scalar value argument.

vmathM3MakeIdentity_V

Construct an identity 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeIdentity_V();
```

Arguments

None

Return Values

The constructed 3x3 matrix

Description

Construct an identity 3x3 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathM3MakeRotationAxis_V

Construct a 3x3 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeRotationAxis_V(
        float radians,
        VmathVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathM3MakeRotationQ_V

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeRotationQ_V(
        VmathQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 3x3 matrix that applies the same rotation as *unitQuat*

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathM3MakeRotationX_V

Construct a 3x3 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeRotationX_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the x axis by the specified radians angle.

vmathM3MakeRotationY_V

Construct a 3x3 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeRotationY_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the y axis by the specified radians angle.

vmathM3MakeRotationZ_V

Construct a 3x3 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeRotationZ_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the z axis by the specified radians angle.

vmathM3MakeRotationZYX_V

Construct a 3x3 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeRotationZYX_V(
        VmathVector3 radiansXYZ
);
```

Arguments

radiansXYZ	3-D vector

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to `rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())`.

vmathM3MakeScale_V

Construct a 3x3 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MakeScale_V(
        VmathVector3 scaleVec
);
```

Arguments

scaleVec 3-D vector

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathM3Mul_V

Multiply two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Mul_V(
        VmathMatrix3 mat0,
        VmathMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

Product of the specified 3x3 matrices

Description

Multiply two 3x3 matrices.

vmathM3MulPerElem_V

Multiply two 3x3 matrices per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3MulPerElem_V(
        VmathMatrix3 mat0,
        VmathMatrix3 mat1
);
```

Arguments

mat0 3x3 matrix
mat1 3x3 matrix

Return Values

3x3 matrix in which each element is the product of the corresponding elements of the specified 3x3 matrices

Description

Multiply two 3x3 matrices element by element.

vmathM3MulV3_V

Multiply a 3x3 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM3MulV3_V(
        VmathMatrix3 mat,
        VmathVector3 vec
);
```

Arguments

mat	3x3 matrix
vec	3-D vector

Return Values

Product of the specified 3x3 matrix and 3-D vector

Description

Multiply a 3x3 matrix by a 3-D vector.

vmathM3Neg_V

Negate all elements of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Neg_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

3x3 matrix containing negated elements of the specified 3x3 matrix

Description

Negate all elements of a 3x3 matrix.

vmathM3PrependScale_V

Prepend (pre-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3PrependScale_V(
        VmathVector3 scaleVec,
        VmathMatrix3 mat
);
```

Arguments

scaleVec	3-D vector
mat	3x3 matrix

Return Values

The product of a scale transformation created from *scaleVec* and *mat*

Description

Pre-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM3Print_V

Print a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3Print_V(
        VmathMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM3Prints_V

Print a 3x3 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3Prints_V(
        VmathMatrix3 mat,
        const char *name
);
```

Arguments

mat	3x3 matrix
name	String printed with the 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix and an associated string identifier. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM3ScalarMul_V

Multiply a 3x3 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3ScalarMul_V(
        VmathMatrix3 mat,
        float scalar
);
```

Arguments

mat	3x3 matrix
scalar	Scalar value

Return Values

Product of the specified 3x3 matrix and scalar

Description

Multiply a 3x3 matrix by a scalar.

vmathM3Select_V

Conditionally select between two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Select_V(
        VmathMatrix3 mat0,
        VmathMatrix3 mat1,
        unsigned int select1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix
select1	False selects the mat0 argument, true selects the mat1 argument

Return Values

Equal to `mat0` if `select1` == 0, or to `mat1` if `select1` != 0

Description

Conditionally select one of the 3x3 matrix arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of `select1` to a VMX register may use more processing time than a branch.

vmathM3SetCol0_V

Set column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3SetCol0_V(
        VmathMatrix3 *result,
        VmathVector3 col0
);
```

Arguments

result	An output 3x3 matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x3 matrix.

vmathM3SetCol1_V

Set column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3SetCol1_V(
        VmathMatrix3 *result,
        VmathVector3 col1
);
```

Arguments

result	An output 3x3 matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x3 matrix.

vmathM3SetCol2_V

Set column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3SetCol2_V(
        VmathMatrix3 *result,
        VmathVector3 col2
);
```

Arguments

result	An output 3x3 matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x3 matrix.

vmathM3SetCol_V

Set the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3SetCol_V(
        VmathMatrix3 *result,
        int col,
        VmathVector3 vec
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the column of a 3x3 matrix referred to by the specified index.

vmathM3SetElem_V

Set the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3SetElem_V(
        VmathMatrix3 *result,
        int col,
        int row,
        float val
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x3 matrix referred to by column and row indices.

vmathM3SetRow_V

Set the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM3SetRow_V(
        VmathMatrix3 *result,
        int row,
        VmathVector3 vec
);
```

Arguments

result	An output 3x3 matrix
row	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the row of a 3x3 matrix referred to by the specified index.

vmathM3Sub_V

Subtract a 3x3 matrix from another 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Sub_V(
        VmathMatrix3 mat0,
        VmathMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

Difference of the specified 3x3 matrices

Description

Subtract a 3x3 matrix from another 3x3 matrix.

vmathM3Transpose_V

Transpose of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM3Transpose_V(
        VmathMatrix3 mat
);
```

Arguments

`mat` 3x3 matrix

Return Values

`mat` transposed

Description

Compute the transpose of a 3x3 matrix.

4x4 Matrix Functions (AoS, by value)

vmathM4AbsPerElem_V

Compute the absolute value of a 4x4 matrix per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4AbsPerElem_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

4x4 matrix in which each element is the absolute value of the corresponding element of the specified 4x4 matrix

Description

Compute the absolute value of each element of a 4x4 matrix.

vmathM4Add_V

Add two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Add_V(
        VmathMatrix4 mat0,
        VmathMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

Sum of the specified 4x4 matrices

Description

Add two 4x4 matrices.

vmathM4AffineInverse_V

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4AffineInverse_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Inverse of the specified 4x4 matrix

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is inverse(M), whose translation vector is -inverse(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions. The result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathM4AppendScale_V

Append (post-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4AppendScale_V(
        VmathMatrix4 mat,
        VmathVector3 scaleVec
);
```

Arguments

mat	4x4 matrix
scaleVec	3-D vector

Return Values

The product of *mat* and a scale transformation created from *scaleVec*

Description

Post-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM4Determinant_V

Determinant of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathM4Determinant_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 4x4 matrix.

vmathM4GetCol0_V

Get column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4GetCol0_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Column 0

Description

Get column 0 of a 4x4 matrix.

vmathM4GetCol1_V

Get column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4GetCol1_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Column 1

Description

Get column 1 of a 4x4 matrix.

vmathM4GetCol2_V

Get column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4GetCol2_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Column 2

Description

Get column 2 of a 4x4 matrix.

vmathM4GetCol3_V

Get column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4GetCol3_V(
        VmathMatrix4 mat
);
```

Arguments

mat	4x4 matrix

Return Values

Column 3

Description

Get column 3 of a 4x4 matrix.

vmathM4GetCol_V

Get the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4GetCol_V(
        VmathMatrix4 mat,
        int col
);
```

Arguments

mat	4x4 matrix
col	Index, expected in the range 0-3

Return Values

The column referred to by the specified index

Description

Get the column of a 4x4 matrix referred to by the specified index.

vmathM4GetElem_V

Get the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathM4GetElem_V(
        VmathMatrix4 mat,
        int col,
        int row
);
```

Arguments

mat	4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3

Return Values

Element selected by *col* and *row*

Description

Get the element of a 4x4 matrix referred to by column and row indices.

vmathM4GetRow_V

Get the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4GetRow_V(
        VmathMatrix4 mat,
        int row
);
```

Arguments

mat	4x4 matrix
row	Index, expected in the range 0-3

Return Values

The row referred to by the specified index

Description

Get the row of a 4x4 matrix referred to by the specified index.

vmathM4GetTranslation_V

Get the translation component of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathM4GetTranslation_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Translation component

Description

Get the translation component of a 4x4 matrix.

vmathM4GetUpper3x3_V

Get the upper-left 3x3 submatrix of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathM4GetUpper3x3_V(
        VmathMatrix4 mat
);
```

Arguments

mat	4x4 matrix

Return Values

Upper-left 3x3 submatrix

Description

Get the upper-left 3x3 submatrix of a 4x4 matrix.

vmathM4Inverse_V

Compute the inverse of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Inverse_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Inverse of *mat*

Description

Compute the inverse of a 4x4 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathM4MakeFromCols_V

Construct a 4x4 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeFromCols_V(
                VmathVector4 col0,
                VmathVector4 col1,
                VmathVector4 col2,
                VmathVector4 col3
);
```

Arguments

col0	4-D vector
col1	4-D vector
col2	4-D vector
col3	4-D vector

Return Values

The 4x4 matrix that contains the specified columns

Description

Construct a 4x4 matrix containing the specified columns.

vmathM4MakeFromM3V3_V

Construct a 4x4 matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeFromM3V3_V(
        VmathMatrix3 mat,
        VmathVector3 translateVec
);
```

Arguments

mat	3x3 matrix
translateVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix whose upper 3x3 elements are equal to the 3x3 matrix argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathM4MakeFromQV3_V

Construct a 4x4 matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeFromQV3_V(
        VmathQuat unitQuat,
        VmathVector3 translateVec
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathM4MakeFromScalar_V

Set all elements of a 4x4 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix with all elements set to the scalar value argument.

vmathM4MakeFromT3_V

Construct a 4x4 matrix from a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeFromT3_V(
        VmathTransform3 mat
);
```

Arguments

mat 3x4 transformation matrix

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix whose upper 3x4 elements are equal to the 3x4 transformation matrix argument and whose bottom row is equal to (0,0,0,1).

vmathM4MakeFrustum_V

Construct a perspective projection matrix based on frustum.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeFrustum_V(
        float left,
        float right,
        float bottom,
        float top,
        float zNear,
        float zFar
);
```

Arguments

left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a perspective projection matrix based on frustum, equal to:

```
2*zNear/(right-left)   0           (right+left)/(right-left)      0
       0       2*zNear/(top-bottom)  (top+bottom)/(top-bottom)      0
       0               0          -(zFar+zNear)/(zFar-zNear)
-2*zFar*zNear/(zFar-zNear)
       0               0                  -1                0  .
```

vmathM4MakeIdentity_V

Construct an identity 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeIdentity_V();
```

Arguments

None

Return Values

The constructed 4x4 matrix

Description

Construct an identity 4x4 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathM4MakeLookAt_V

Construct viewing matrix based on eye position, position looked at, and up direction.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeLookAt_V(
        VmathPoint3 eyePos,
        VmathPoint3 lookAtPos,
        VmathVector3 upVec
);
```

Arguments

eyePos	3-D point
lookAtPos	3-D point
upVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct the inverse of a coordinate frame that is centered at the eye position, with z axis directed away from lookAtPos, and y axis oriented to best match the up direction.

vmathM4MakeOrthographic_V

Construct an orthographic projection matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeOrthographic_V(
        float left,
        float right,
        float bottom,
        float top,
        float zNear,
        float zFar
);
```

Arguments

left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct an orthographic projection matrix, equal to

```
  2/(right-left)        0              0        -(right+left)/(right-left)
      0          2/(top-bottom)        0        -(top+bottom)/(top-bottom)
      0                 0       -2/(zFar-zNear) -(zFar+zNear)/(zFar-zNear)
      0                 0              0                   1  .
```

vmathM4MakePerspective_V

Construct a perspective projection matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakePerspective_V(
        float fovyRadians,
        float aspect,
        float zNear,
        float zFar
);
```

Arguments

fovyRadians	Scalar value
aspect	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a perspective projection matrix, equal to:

```
cot(fovyRadians/2)/aspect   0                  0                  0
       0             cot(fovyRadians/2)        0                  0
       0                    0   (zFar+zNear)/(zNear-zFar)
2*zFar*zNear/(zNear-zFar)
       0                    0             -1                 0  .
```

vmathM4MakeRotationAxis_V

Construct a 4x4 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeRotationAxis_V(
        float radians,
        VmathVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathM4MakeRotationQ_V

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeRotationQ_V(
        VmathQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 4x4 matrix that applies the same rotation as *unitQuat*

Description

Construct a 4x4 matrix that applies the same rotation as the specified unit-length quaternion.

vmathM4MakeRotationX_V

Construct a 4x4 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeRotationX_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the x axis by the specified radians angle.

vmathM4MakeRotationY_V

Construct a 4x4 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeRotationY_V(
        float radians
);
```

Arguments

radians	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the y axis by the specified radians angle.

vmathM4MakeRotationZ_V

Construct a 4x4 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeRotationZ_V(
        float radians
);
```

Arguments

radians	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the z axis by the specified radians angle.

vmathM4MakeRotationZYX_V

Construct a 4x4 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeRotationZYX_V(
        VmathVector3 radiansXYZ
);
```

Arguments

radiansXYZ 3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to *rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())*.

vmathM4MakeScale_V

Construct a 4x4 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeScale_V(
        VmathVector3 scaleVec
);
```

Arguments

scaleVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathM4MakeTranslation_V

Construct a 4x4 matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MakeTranslation_V(
        VmathVector3 translateVec
);
```

Arguments

translateVec 3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathM4Mul_V

Multiply two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Mul_V(
        VmathMatrix4 mat0,
        VmathMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

Product of the specified 4x4 matrices

Description

Multiply two 4x4 matrices.

vmathM4MulP3_V

Multiply a 4x4 matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4MulP3_V(
        VmathMatrix4 mat,
        VmathPoint3 pnt
);
```

Arguments

mat	4x4 matrix
pnt	3-D point

Return Values

Product of the specified 4x4 matrix and 3-D point

Description

Multiply a 4x4 matrix by a 3-D point treated as if it were a 4-D vector with the w element equal to 1.

vmathM4MulPerElem_V

Multiply two 4x4 matrices per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MulPerElem_V(
        VmathMatrix4 mat0,
        VmathMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

4x4 matrix in which each element is the product of the corresponding elements of the specified 4x4 matrices

Description

Multiply two 4x4 matrices element by element.

vmathM4MulT3_V

Multiply a 4x4 matrix by a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4MulT3_V(
        VmathMatrix4 mat,
        VmathTransform3 tfrm
);
```

Arguments

mat	4x4 matrix
tfrm	3x4 transformation matrix

Return Values

Product of the specified 4x4 matrix and 3x4 transformation matrix

Description

Multiply a 4x4 matrix by a 3x4 transformation matrix treated as if it were a 4x4 matrix with the bottom row equal to (0,0,0,1).

vmathM4MulV3_V

Multiply a 4x4 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4MulV3_V(
        VmathMatrix4 mat,
        VmathVector3 vec
);
```

Arguments

mat	4x4 matrix
vec	3-D vector

Return Values

Product of the specified 4x4 matrix and 3-D vector

Description

Multiply a 4x4 matrix by a 3-D vector treated as if it were a 4-D vector with the w element equal to 0.

vmathM4MulV4_V

Multiply a 4x4 matrix by a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathM4MulV4_V(
        VmathMatrix4 mat,
        VmathVector4 vec
);
```

Arguments

mat	4x4 matrix
vec	4-D vector

Return Values

Product of the specified 4x4 matrix and 4-D vector

Description

Multiply a 4x4 matrix by a 4-D vector.

vmathM4Neg_V

Negate all elements of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Neg_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

4x4 matrix containing negated elements of the specified 4x4 matrix

Description

Negate all elements of a 4x4 matrix.

vmathM4OrthoInverse_V

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix with an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4OrthoInverse_V(
        VmathMatrix4 mat
);
```

Arguments

mat	4x4 matrix

Return Values

Inverse of the specified 4x4 matrix

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), whose translation vector is -transpose(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions.

vmathM4PrependScale_V

Prepend (pre-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4PrependScale_V(
        VmathVector3 scaleVec,
        VmathMatrix4 mat
);
```

Arguments

scaleVec	3-D vector
mat	4x4 matrix

Return Values

The product of a scale transformation created from *scaleVec* and *mat*

Description

Pre-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathM4Print_V

Print a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4Print_V(
        VmathMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM4Prints_V

Print a 4x4 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4Prints_V(
        VmathMatrix4 mat,
        const char *name
);
```

Arguments

mat	4x4 matrix
name	String printed with the 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix and an associated string identifier. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathM4ScalarMul_V

Multiply a 4x4 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4ScalarMul_V(
        VmathMatrix4 mat,
        float scalar
);
```

Arguments

mat	4x4 matrix
scalar	Scalar value

Return Values

Product of the specified 4x4 matrix and scalar

Description

Multiply a 4x4 matrix by a scalar.

vmathM4Select_V

Conditionally select between two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Select_V(
        VmathMatrix4 mat0,
        VmathMatrix4 mat1,
        unsigned int select1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix
select1	False selects the mat0 argument, true selects the mat1 argument

Return Values

Equal to `mat0` if `select1` == 0, or to `mat1` if `select1` != 0

Description

Conditionally select one of the 4x4 matrix arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of `select1` to a VMX register may use more processing time than a branch.

vmathM4SetCol0_V

Set column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetCol0_V(
        VmathMatrix4 *result,
        VmathVector4 col0
);
```

Arguments

result	An output 4x4 matrix
col0	4-D vector

Return Values

None

Description

Set column 0 of a 4x4 matrix.

vmathM4SetCol1_V

Set column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetCol1_V(
        VmathMatrix4 *result,
        VmathVector4 col1
);
```

Arguments

result	An output 4x4 matrix
col1	4-D vector

Return Values

None

Description

Set column 1 of a 4x4 matrix.

vmathM4SetCol2_V

Set column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetCol2_V(
        VmathMatrix4 *result,
        VmathVector4 col2
);
```

Arguments

result	An output 4x4 matrix
col2	4-D vector

Return Values

None

Description

Set column 2 of a 4x4 matrix.

vmathM4SetCol3_V

Set column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetCol3_V(
        VmathMatrix4 *result,
        VmathVector4 col3
);
```

Arguments

result	An output 4x4 matrix
col3	4-D vector

Return Values

None

Description

Set column 3 of a 4x4 matrix.

vmathM4SetCol_V

Set the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetCol_V(
        VmathMatrix4 *result,
        int col,
        VmathVector4 vec
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the column of a 4x4 matrix referred to by the specified index.

vmathM4SetElem_V

Set the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetElem_V(
        VmathMatrix4 *result,
        int col,
        int row,
        float val
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3
val	Scalar value

Return Values

None

Description

Set the element of a 4x4 matrix referred to by column and row indices.

vmathM4SetRow_V

Set the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetRow_V(
        VmathMatrix4 *result,
        int row,
        VmathVector4 vec
);
```

Arguments

result	An output 4x4 matrix
row	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the row of a 4x4 matrix referred to by the specified index.

vmathM4SetTranslation_V

Set translation component.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetTranslation_V(
        VmathMatrix4 *result,
        VmathVector3 translateVec
);
```

Arguments

result	An output 4x4 matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 4x4 matrix equal to the specified 3-D vector.

Notes

This function does not change the bottom row elements.

vmathM4SetUpper3x3_V

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathM4SetUpper3x3_V(
        VmathMatrix4 *result,
        VmathMatrix3 mat3
);
```

Arguments

result	An output 4x4 matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 4x4 matrix equal to the specified 3x3 matrix.

Notes

This function does not change the bottom row elements.

vmathM4Sub_V

Subtract a 4x4 matrix from another 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Sub_V(
        VmathMatrix4 mat0,
        VmathMatrix4 mat1
);
```

Arguments

mat0 4x4 matrix
mat1 4x4 matrix

Return Values

Difference of the specified 4x4 matrices

Description

Subtract a 4x4 matrix from another 4x4 matrix.

vmathM4Transpose_V

Transpose of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix4 vmathM4Transpose_V(
        VmathMatrix4 mat
);
```

Arguments

mat	4x4 matrix

Return Values

mat transposed

Description

Compute the transpose of a 4x4 matrix.

Transformation Functions (AoS, by value)

vmathT3AbsPerElem_V

Compute the absolute value of a 3x4 transformation matrix per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3AbsPerElem_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

3x4 transformation matrix in which each element is the absolute value of the corresponding element of the specified 3x4 transformation matrix

Description

Compute the absolute value of each element of a 3x4 transformation matrix.

vmathT3AppendScale_V

Append (post-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3AppendScale_V(
        VmathTransform3 tfrm,
        VmathVector3 scaleVec
);
```

Arguments

tfrm	3x4 transformation matrix
scaleVec	3-D vector

Return Values

The product of *tfrm* and a scale transformation created from *scaleVec*

Description

Post-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathT3GetCol0_V

Get column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3GetCol0_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 0

Description

Get column 0 of a 3x4 transformation matrix.

vmathT3GetCol1_V

Get column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3GetCol1_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 1

Description

Get column 1 of a 3x4 transformation matrix.

vmathT3GetCol2_V

Get column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3GetCol2_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 2

Description

Get column 2 of a 3x4 transformation matrix.

vmathT3GetCol3_V

Get column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3GetCol3_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 3

Description

Get column 3 of a 3x4 transformation matrix.

vmathT3GetCol_V

Get the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3GetCol_V(
        VmathTransform3 tfrm,
        int col
);
```

Arguments

tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3

Return Values

The column referred to by the specified index

Description

Get the column of a 3x4 transformation matrix referred to by the specified index.

vmathT3GetElem_V

Get the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline float vmathT3GetElem_V(
        VmathTransform3 tfrm,
        int col,
        int row
);
```

Arguments

tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x4 transformation matrix referred to by column and row indices.

vmathT3GetRow_V

Get the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector4 vmathT3GetRow_V(
        VmathTransform3 tfrm,
        int row
);
```

Arguments

tfrm 3x4 transformation matrix
row Index, expected in the range 0-2

Return Values

The row referred to by the specified index

Description

Get the row of a 3x4 transformation matrix referred to by the specified index.

vmathT3GetTranslation_V

Get the translation component of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3GetTranslation_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Translation component

Description

Get the translation component of a 3x4 transformation matrix.

vmathT3GetUpper3x3_V

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathMatrix3 vmathT3GetUpper3x3_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Upper-left 3x3 submatrix

Description

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

vmathT3Inverse_V

Inverse of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3Inverse_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Inverse of *tfrm*

Description

Compute the inverse of a 3x4 transformation matrix.

Notes

Result is unpredictable when the determinant of the left 3x3 submatrix is equal to or near 0.

vmathT3MakeFromCols_V

Construct a 3x4 transformation matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeFromCols_V(
        VmathVector3 col0,
        VmathVector3 col1,
        VmathVector3 col2,
        VmathVector3 col3
);
```

Arguments

col0	3-D vector
col1	3-D vector
col2	3-D vector
col3	3-D vector

Return Values

The 3x4 transformation matrix that contains the specified columns

Description

Construct a 3x4 transformation matrix containing the specified columns.

vmathT3MakeFromM3V3_V

Construct a 3x4 transformation matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeFromM3V3_V(
        VmathMatrix3 tfrm,
        VmathVector3 translateVec
);
```

Arguments

tfrm	3x3 matrix
translateVec	3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix whose upper 3x3 elements are equal to the 3x3 matrix argument and whose translation component is equal to the 3-D vector argument.

vmathT3MakeFromQV3_V

Construct a 3x4 transformation matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeFromQV3_V(
        VmathQuat unitQuat,
        VmathVector3 translateVec
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument and whose translation component is equal to the 3-D vector argument.

vmathT3MakeFromScalar_V

Set all elements of a 3x4 transformation matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeFromScalar_V(
        float scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix with all elements set to the scalar value argument.

vmathT3MakeIdentity_V

Construct an identity 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeIdentity_V();
```

Arguments

None

Return Values

The constructed 3x4 transformation matrix

Description

Construct an identity 3x4 transformation matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathT3MakeRotationAxis_V

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeRotationAxis_V(
        float radians,
        VmathVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathT3MakeRotationQ_V

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeRotationQ_V(
        VmathQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 3x4 transformation matrix that applies the same rotation as *unitQuat*

Description

Construct a 3x4 transformation matrix that applies the same rotation as the specified unit-length quaternion.

vmathT3MakeRotationX_V

Construct a 3x4 transformation matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeRotationX_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the x axis by the specified radians angle.

vmathT3MakeRotationY_V

Construct a 3x4 transformation matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeRotationY_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the y axis by the specified radians angle.

vmathT3MakeRotationZ_V

Construct a 3x4 transformation matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeRotationZ_V(
        float radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the z axis by the specified radians angle.

vmathT3MakeRotationZYX_V

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeRotationZYX_V(
        VmathVector3 radiansXYZ
);
```

Arguments

radiansXYZ 3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to `rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())`.

vmathT3MakeScale_V

Construct a 3x4 transformation matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeScale_V(
        VmathVector3 scaleVec
);
```

Arguments

scaleVec	3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathT3MakeTranslation_V

Construct a 3x4 transformation matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MakeTranslation_V(
        VmathVector3 translateVec
);
```

Arguments

translateVec	3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathT3Mul_V

Multiply two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3Mul_V(
        VmathTransform3 tfrm0,
        VmathTransform3 tfrm1
);
```

Arguments

tfrm0 3x4 transformation matrix
tfrm1 3x4 transformation matrix

Return Values

Product of the specified 3x4 transformation matrices

Description

Multiply two 3x4 transformation matrices.

vmathT3MulP3_V

Multiply a 3x4 transformation matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathPoint3 vmathT3MulP3_V(
        VmathTransform3 tfrm,
        VmathPoint3 pnt
);
```

Arguments

tfrm	3x4 transformation matrix
pnt	3-D point

Return Values

Product of the specified 3x4 transformation matrix and 3-D point

Description

Applies the 3x3 upper-left submatrix and the translation component of a 3x4 transformation matrix to a 3-D point.

vmathT3MulPerElem_V

Multiply two 3x4 transformation matrices per element.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3MulPerElem_V(
        VmathTransform3 tfrm0,
        VmathTransform3 tfrm1
);
```

Arguments

tfrm0 3x4 transformation matrix
tfrm1 3x4 transformation matrix

Return Values

3x4 transformation matrix in which each element is the product of the corresponding elements of the specified 3x4 transformation matrices

Description

Multiply two 3x4 transformation matrices element by element.

vmathT3MulV3_V

Multiply a 3x4 transformation matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathVector3 vmathT3MulV3_V(
        VmathTransform3 tfrm,
        VmathVector3 vec
);
```

Arguments

tfrm	3x4 transformation matrix
vec	3-D vector

Return Values

Product of the specified 3x4 transformation matrix and 3-D vector

Description

Applies the 3x3 upper-left submatrix (but not the translation component) of a 3x4 transformation matrix to a 3-D vector.

vmathT3OrthoInverse_V

Compute the inverse of a 3x4 transformation matrix, expected to have an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3OrthoInverse_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Inverse of the specified 3x4 transformation matrix

Description

Naming the upper-left 3x3 submatrix of the specified 3x4 transformation matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), and whose translation vector is -transpose(M)*v.

Notes

This can be used to achieve better performance than a general inverse when the specified 3x4 transformation matrix meets the given restrictions.

vmathT3PrependScale_V

Prepend (pre-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3PrependScale_V(
        VmathVector3 scaleVec,
        VmathTransform3 tfrm
);
```

Arguments

scaleVec	3-D vector
tfrm	3x4 transformation matrix

Return Values

The product of a scale transformation created from *scaleVec* and *tfrm*

Description

Pre-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathT3Print_V

Print a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3Print_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathT3Prints_V

Print a 3x4 transformation matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3Prints_V(
        VmathTransform3 tfrm,
        const char *name
);
```

Arguments

tfrm	3x4 transformation matrix
name	String printed with the 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix and an associated string identifier. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathT3Select_V

Conditionally select between two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline VmathTransform3 vmathT3Select_V(
        VmathTransform3 tfrm0,
        VmathTransform3 tfrm1,
        unsigned int select1
);
```

Arguments

tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix
select1	False selects the tfrm0 argument, true selects the tfrm1 argument

Return Values

Equal to `tfrm0` if `select1` == 0, or to `tfrm1` if `select1` != 0

Description

Conditionally select one of the 3x4 transformation matrix arguments.

Notes

This function uses a conditional select instruction to avoid a branch. However, the transfer of `select1` to a VMX register may use more processing time than a branch.

vmathT3SetCol0_V

Set column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetCol0_V(
        VmathTransform3 *result,
        VmathVector3 col0
);
```

Arguments

result	An output 3x4 transformation matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x4 transformation matrix.

vmathT3SetCol1_V

Set column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetCol1_V(
        VmathTransform3 *result,
        VmathVector3 col1
);
```

Arguments

result	An output 3x4 transformation matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x4 transformation matrix.

vmathT3SetCol2_V

Set column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetCol2_V(
        VmathTransform3 *result,
        VmathVector3 col2
);
```

Arguments

result	An output 3x4 transformation matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x4 transformation matrix.

vmathT3SetCol3_V

Set column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetCol3_V(
        VmathTransform3 *result,
        VmathVector3 col3
);
```

Arguments

result	An output 3x4 transformation matrix
col3	3-D vector

Return Values

None

Description

Set column 3 of a 3x4 transformation matrix.

vmathT3SetCol_V

Set the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetCol_V(
        VmathTransform3 *result,
        int col,
        VmathVector3 vec
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
vec	3-D vector

Return Values

None

Description

Set the column of a 3x4 transformation matrix referred to by the specified index.

vmathT3SetElem_V

Set the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetElem_V(
        VmathTransform3 *result,
        int col,
        int row,
        float val
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x4 transformation matrix referred to by column and row indices.

vmathT3SetRow_V

Set the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetRow_V(
        VmathTransform3 *result,
        int row,
        VmathVector4 vec
);
```

Arguments

result	An output 3x4 transformation matrix
row	Index, expected in the range 0-2
vec	4-D vector

Return Values

None

Description

Set the row of a 3x4 transformation matrix referred to by the specified index.

vmathT3SetTranslation_V

Set translation component.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetTranslation_V(
        VmathTransform3 *result,
        VmathVector3 translateVec
);
```

Arguments

result	An output 3x4 transformation matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 3x4 transformation matrix equal to the specified 3-D vector.

vmathT3SetUpper3x3_V

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_aos_v.h>
static inline void vmathT3SetUpper3x3_V(
        VmathTransform3 *result,
        VmathMatrix3 mat3
);
```

Arguments

result	An output 3x4 transformation matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 3x4 transformation matrix equal to the specified 3x3 matrix.

3-D Vector Functions (SoA, by value)

vmathSoaV3AbsPerElem_V

Compute the absolute value of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3AbsPerElem_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the absolute value of the corresponding element of vec

Description

Compute the absolute value of each element of a 3-D vector.

vmathSoaV3Add_V

Add two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Add_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Sum of the specified 3-D vectors

Description

Add two 3-D vectors.

vmathSoaV3AddP3_V

Add a 3-D vector to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaV3AddP3_V(
        VmathSoaVector3 vec,
        VmathSoaPoint3 pnt
);
```

Arguments

vec 3-D vector
pnt 3-D point

Return Values

Sum of the specified 3-D vector and 3-D point

Description

Add a 3-D vector to a 3-D point.

vmathSoaV3CopySignPerElem_V

Copy sign from one 3-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3CopySignPerElem_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathSoaV3Cross_V

Compute cross product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Cross_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Cross product of the specified 3-D vectors

Description

Compute cross product of two 3-D vectors.

vmathSoaV3CrossMatrix_V

Cross-product matrix of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaV3CrossMatrix_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Cross-product matrix of *vec*

Description

Compute a matrix that, when multiplied by a 3-D vector, produces the same result as a cross product with that 3-D vector.

vmathSoaV3CrossMatrixMul_V

Create cross-product matrix and multiply.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaV3CrossMatrixMul_V(
        VmathSoaVector3 vec,
        VmathSoaMatrix3 mat
);
```

Arguments

vec	3-D vector
mat	3x3 matrix

Return Values

Product of cross-product matrix of *vec* and *mat*

Description

Multiply a cross-product matrix by another matrix.

Notes

Faster than separately creating a cross-product matrix and multiplying.

vmathSoaV3DivPerElem_V

Divide two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3DivPerElem_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the quotient of the corresponding elements of the specified 3-D vectors

Description

Divide two 3-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathSoaV3Dot_V

Compute the dot product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3Dot_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Dot product of the specified 3-D vectors

Description

Compute the dot product of two 3-D vectors.

vmathSoaV3Get4Aos_V

Extract four AoS 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3Get4Aos_V(
        VmathSoaVector3 vec,
        VmathVector3 *result0,
        VmathVector3 *result1,
        VmathVector3 *result2,
        VmathVector3 *result3
);
```

Arguments

vec	3-D vector
result0	An output AoS 3-D vector
result1	An output AoS 3-D vector
result2	An output AoS 3-D vector
result3	An output AoS 3-D vector

Return Values

None

Description

Extract four AoS 3-D vectors from four slots of an SoA 3-D vector (transpose the data format).

vmathSoaV3GetElem_V

Get an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3GetElem_V(
        VmathSoaVector3 vec,
        int idx
);
```

Arguments

vec	3-D vector
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathSoaV3GetX_V

Get the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3GetX_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

x element of a 3-D vector

Description

Get the x element of a 3-D vector.

vmathSoaV3GetY_V

Get the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3GetY_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

y element of a 3-D vector

Description

Get the y element of a 3-D vector.

vmathSoaV3GetZ_V

Get the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3GetZ_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

z element of a 3-D vector

Description

Get the z element of a 3-D vector.

vmathSoaV3Length_V

Compute the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3Length_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Length of the specified 3-D vector

Description

Compute the length of a 3-D vector.

vmathSoaV3LengthSqr_V

Compute the square of the length of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3LengthSqr_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Square of the length of the specified 3-D vector

Description

Compute the square of the length of a 3-D vector.

vmathSoaV3Lerp_V

Linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Lerp_V(
        vec_float4 t,
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

t	Interpolation parameter
vec0	3-D vector
vec1	3-D vector

Return Values

Interpolated 3-D vector

Description

Linearly interpolate between two 3-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaV3LoadXYZArray_V

Load four three-float 3-D vectors, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3LoadXYZArray_V(
        VmathSoaVector3 *vec,
        const vec_float4 *threeQuads
);
```

Arguments

vec	An output 3-D vector
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D vectors, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four slots of an SoA 3-D vector.

vmathSoaV3MakeFrom4Aos_V

Insert four AoS 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeFrom4Aos_V(
        VmathVector3 vec0,
        VmathVector3 vec1,
        VmathVector3 vec2,
        VmathVector3 vec3
);
```

Arguments

vec0	AoS 3-D vector
vec1	AoS 3-D vector
vec2	AoS 3-D vector
vec3	AoS 3-D vector

Return Values

The constructed SoA 3-D vector

Description

Insert four AoS 3-D vectors into four slots of an SoA 3-D vector (transpose the data format).

vmathSoaV3MakeFromAos_V

Replicate an AoS 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeFromAos_V(
        VmathVector3 vec
);
```

Arguments

vec AoS 3-D vector

Return Values

The constructed SoA 3-D vector

Description

Replicate an AoS 3-D vector in all four slots of an SoA 3-D vector.

vmathSoaV3MakeFromElems_V

Construct a 3-D vector from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeFromElems_V(
        vec_float4 x,
        vec_float4 y,
        vec_float4 z
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value

Return Values

The 3-D vector that contains the specified elements

Description

Construct a 3-D vector containing the specified x, y, and z elements.

vmathSoaV3MakeFromP3_V

Copy elements from a 3-D point into a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeFromP3_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector containing the x, y, and z elements of the specified 3-D point.

vmathSoaV3MakeFromScalar_V

Set all elements of a 3-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar	Scalar value

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector with all elements set to the scalar value argument.

vmathSoaV3MakeXAxis_V

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeXAxis_V();
```

Arguments

None

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector equal to (1,0,0).

vmathSoaV3MakeYAxis_V

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeYAxis_V();
```

Arguments

None

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector equal to (0,1,0).

vmathSoaV3MakeZAxis_V

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MakeZAxis_V();
```

Arguments

None

Return Values

The constructed 3-D vector

Description

Construct a 3-D vector equal to (0,0,1).

vmathSoaV3MaxElem_V

Maximum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3MaxElem_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 3-D vector.

vmathSoaV3MaxPerElem_V

Maximum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MaxPerElem_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors

Description

Create a 3-D vector in which each element is the maximum of the corresponding elements of the specified 3-D vectors.

vmathSoaV3MinElem_V

Minimum element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3MinElem_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 3-D vector.

vmathSoaV3MinPerElem_V

Minimum of two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MinPerElem_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the minimum of the corresponding elements of the specified 3-D vectors

Description

Create a 3-D vector in which each element is the minimum of the corresponding elements of two specified 3-D vectors.

vmathSoaV3MulPerElem_V

Multiply two 3-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3MulPerElem_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

3-D vector in which each element is the product of the corresponding elements of the specified 3-D vectors

Description

Multiply two 3-D vectors element by element.

vmathSoaV3Neg_V

Negate all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Neg_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector containing negated elements of the specified 3-D vector

Description

Negate all elements of a 3-D vector.

vmathSoaV3Normalize_V

Normalize a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Normalize_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

The specified 3-D vector scaled to unit length

Description

Compute a normalized 3-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathSoaV3Outer_V

Outer product of two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaV3Outer_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

The 3x3 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*

Description

Compute the outer product of two 3-D vectors.

vmathSoaV3Print_V

Print a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3Print_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

None

Description

Print a 3-D vector. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV3Prints_V

Print a 3-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3Prints_V(
        VmathSoaVector3 vec,
        const char *name
);
```

Arguments

vec	3-D vector
name	String printed with the 3-D vector

Return Values

None

Description

Print a 3-D vector and an associated string identifier. Prints the 3-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV3RecipPerElem_V

Compute the reciprocal of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3RecipPerElem_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector

Description

Create a 3-D vector in which each element is the reciprocal of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathSoaV3RowMul_V

Pre-multiply a row vector by a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3RowMul_V(
        VmathSoaVector3 vec,
        VmathSoaMatrix3 mat
);
```

Arguments

vec	3-D vector
mat	3x3 matrix

Return Values

Product of a row-vector and a 3x3 matrix

Description

Transpose a 3-D vector into a row vector and pre-multiply by 3x3 matrix.

vmathSoaV3RsqrtPerElem_V

Compute the reciprocal square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3RsqrtPerElem_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector

Description

Create a 3-D vector in which each element is the reciprocal square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathSoaV3ScalarDiv_V

Divide a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3ScalarDiv_V(
        VmathSoaVector3 vec,
        vec_float4 scalar
);
```

Arguments

vec	3-D vector
scalar	Scalar value

Return Values

Quotient of the specified 3-D vector and scalar

Description

Divide a 3-D vector by a scalar.

vmathSoaV3ScalarMul_V

Multiply a 3-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3ScalarMul_V(
        VmathSoaVector3 vec,
        vec_float4 scalar
);
```

Arguments

vec	3-D vector
scalar	Scalar value

Return Values

Product of the specified 3-D vector and scalar

Description

Multiply a 3-D vector by a scalar.

vmathSoaV3Select_V

Conditionally select between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Select_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1,
        vec_uint4 select1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
select1	For each of the four word slots, this mask selects either the 3-D vector in the corresponding slot of *vec0* or the 3-D vector in the corresponding slot of *vec1*. A 0 bit selects from *vec0* whereas a 1 bit selects from *vec1*. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the 3-D vector at the corresponding slot of *vec0* or *vec1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *vec0*, and a value of 0xFFFFFFFF selects the slot of *vec1*

Description

Conditionally select one of the 3-D vectors at each of the corresponding slots of *vec0* or *vec1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaV3SetElem_V

Set an x, y, or z element of a 3-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3SetElem_V(
        VmathSoaVector3 *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output 3-D vector
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D vector by specifying an index of 0, 1, or 2, respectively.

vmathSoaV3SetX_V

Set the x element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3SetX_V(
        VmathSoaVector3 *result,
        vec_float4 x
);
```

Arguments

result	An output 3-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D vector to the specified scalar value.

vmathSoaV3SetY_V

Set the y element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3SetY_V(
        VmathSoaVector3 *result,
        vec_float4 y
);
```

Arguments

result	An output 3-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D vector to the specified scalar value.

vmathSoaV3SetZ_V

Set the z element of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3SetZ_V(
        VmathSoaVector3 *result,
        vec_float4 z
);
```

Arguments

result	An output 3-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D vector to the specified scalar value.

vmathSoaV3Slerp_V

Spherical linear interpolation between two 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Slerp_V(
        vec_float4 t,
        VmathSoaVector3 unitVec0,
        VmathSoaVector3 unitVec1
);
```

Arguments

t	Interpolation parameter
unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

Interpolated 3-D vector

Description

Perform spherical linear interpolation between two 3-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp *t* between 0 and 1.

vmathSoaV3SqrtPerElem_V

Compute the square root of a 3-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3SqrtPerElem_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector

Description

Create a 3-D vector in which each element is the square root of the corresponding element of the specified 3-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathSoaV3StoreHalfFloats_V

Store eight slots of two SoA 3-D vectors as half-floats.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3StoreHalfFloats_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1,
        vec_ushort8 *threeQuads
);
```

Arguments

vec0	3-D vector
vec1	3-D vector
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight slots of two SoA 3-D vectors in three quadwords of half-float values. Numbering slots of *vec0* as 0..3 and slots of *vec1* as 4..7, the output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathSoaV3StoreXYZArray_V

Store four slots of an SoA 3-D vector in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV3StoreXYZArray_V(
        VmathSoaVector3 vec,
        vec_float4 *threeQuads
);
```

Arguments

vec	3-D vector
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four slots of an SoA 3-D vector in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathSoaV3Sub_V

Subtract a 3-D vector from another 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV3Sub_V(
        VmathSoaVector3 vec0,
        VmathSoaVector3 vec1
);
```

Arguments

vec0	3-D vector
vec1	3-D vector

Return Values

Difference of the specified 3-D vectors

Description

Subtract a 3-D vector from another 3-D vector.

vmathSoaV3Sum_V

Compute the sum of all elements of a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV3Sum_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 3-D vector.

4-D Vector Functions (SoA, by value)

vmathSoaV4AbsPerElem_V

Compute the absolute value of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4AbsPerElem_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the absolute value of the corresponding element of vec

Description

Compute the absolute value of each element of a 4-D vector.

vmathSoaV4Add_V

Add two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Add_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Sum of the specified 4-D vectors

Description

Add two 4-D vectors.

vmathSoaV4CopySignPerElem_V

Copy sign from one 4-D vector to another, per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4CopySignPerElem_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element has the magnitude of the corresponding element of *vec0* and the sign of the corresponding element of *vec1*

Description

For each element, create a value composed of the magnitude of *vec0* and the sign of *vec1*.

vmathSoaV4DivPerElem_V

Divide two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4DivPerElem_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the quotient of the corresponding elements of the specified 4-D vectors

Description

Divide two 4-D vectors element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathSoaV4Dot_V

Compute the dot product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4Dot_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Dot product of the specified 4-D vectors

Description

Compute the dot product of two 4-D vectors.

vmathSoaV4Get4Aos_V

Extract four AoS 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4Get4Aos_V(
        VmathSoaVector4 vec,
        VmathVector4 *result0,
        VmathVector4 *result1,
        VmathVector4 *result2,
        VmathVector4 *result3
);
```

Arguments

vec	4-D vector
result0	An output AoS 4-D vector
result1	An output AoS 4-D vector
result2	An output AoS 4-D vector
result3	An output AoS 4-D vector

Return Values

None

Description

Extract four AoS 4-D vectors from four slots of an SoA 4-D vector (transpose the data format).

vmathSoaV4GetElem_V

Get an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4GetElem_V(
        VmathSoaVector4 vec,
        int idx
);
```

Arguments

vec	4-D vector
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaV4GetW_V

Get the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4GetW_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

w element of a 4-D vector

Description

Get the w element of a 4-D vector.

vmathSoaV4GetX_V

Get the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4GetX_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

x element of a 4-D vector

Description

Get the x element of a 4-D vector.

vmathSoaV4GetXYZ_V

Get the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaV4GetXYZ_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

3-D vector containing x, y, and z elements

Description

Extract a 4-D vector's x, y, and z elements into a 3-D vector.

vmathSoaV4GetY_V

Get the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4GetY_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

y element of a 4-D vector

Description

Get the y element of a 4-D vector.

vmathSoaV4GetZ_V

Get the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4GetZ_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

z element of a 4-D vector

Description

Get the z element of a 4-D vector.

vmathSoaV4Length_V

Compute the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4Length_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Length of the specified 4-D vector

Description

Compute the length of a 4-D vector.

vmathSoaV4LengthSqr_V

Compute the square of the length of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4LengthSqr_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Square of the length of the specified 4-D vector

Description

Compute the square of the length of a 4-D vector.

vmathSoaV4Lerp_V

Linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Lerp_V(
        vec_float4 t,
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

t	Interpolation parameter
vec0	4-D vector
vec1	4-D vector

Return Values

Interpolated 4-D vector

Description

Linearly interpolate between two 4-D vectors.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaV4MakeFrom4Aos_V

Insert four AoS 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFrom4Aos_V(
        VmathVector4 vec0,
        VmathVector4 vec1,
        VmathVector4 vec2,
        VmathVector4 vec3
);
```

Arguments

vec0	AoS 4-D vector
vec1	AoS 4-D vector
vec2	AoS 4-D vector
vec3	AoS 4-D vector

Return Values

The constructed SoA 4-D vector

Description

Insert four AoS 4-D vectors into four slots of an SoA 4-D vector (transpose the data format).

vmathSoaV4MakeFromAos_V

Replicate an AoS 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromAos_V(
        VmathVector4 vec
);
```

Arguments

vec AoS 4-D vector

Return Values

The constructed SoA 4-D vector

Description

Replicate an AoS 4-D vector in all four slots of an SoA 4-D vector.

vmathSoaV4MakeFromElems_V

Construct a 4-D vector from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromElems_V(
        vec_float4 x,
        vec_float4 y,
        vec_float4 z,
        vec_float4 w
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value
w Scalar value

Return Values

The 4-D vector that contains the specified elements

Description

Construct a 4-D vector containing the specified x, y, z, and w elements.

vmathSoaV4MakeFromP3_V

Copy x, y, and z from a 3-D point into a 4-D vector, and set w to 1.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromP3_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D point and with the w element set to 1.

vmathSoaV4MakeFromQ_V

Copy elements from a quaternion into a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromQ_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector containing the x, y, z, and w elements of the specified quaternion.

vmathSoaV4MakeFromScalar_V

Set all elements of a 4-D vector to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with all elements set to the scalar value argument.

vmathSoaV4MakeFromV3_V

Copy x, y, and z from a 3-D vector into a 4-D vector, and set w to 0.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromV3_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to 0.

vmathSoaV4MakeFromV3Scalar_V

Construct a 4-D vector from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeFromV3Scalar_V(
        VmathSoaVector3 xyz,
        vec_float4 w
);
```

Arguments

xyz	3-D vector
w	Scalar value

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathSoaV4MakeWAxis_V

Construct w axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeWAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (0,0,0,1).

vmathSoaV4MakeXAxis_V

Construct x axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeXAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (1,0,0,0).

vmathSoaV4MakeYAxis_V

Construct y axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeYAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (0,1,0,0).

vmathSoaV4MakeZAxis_V

Construct z axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MakeZAxis_V();
```

Arguments

None

Return Values

The constructed 4-D vector

Description

Construct a 4-D vector equal to (0,0,1,0).

vmathSoaV4MaxElem_V

Maximum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4MaxElem_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Maximum value of all elements of *vec*

Description

Compute the maximum value of all elements of a 4-D vector.

vmathSoaV4MaxPerElem_V

Maximum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MaxPerElem_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors

Description

Create a 4-D vector in which each element is the maximum of the corresponding elements of the specified 4-D vectors.

vmathSoaV4MinElem_V

Minimum element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4MinElem_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Minimum value of all elements of *vec*

Description

Compute the minimum value of all elements of a 4-D vector.

vmathSoaV4MinPerElem_V

Minimum of two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MinPerElem_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the minimum of the corresponding elements of the specified 4-D vectors

Description

Create a 4-D vector in which each element is the minimum of the corresponding elements of two specified 4-D vectors.

vmathSoaV4MulPerElem_V

Multiply two 4-D vectors per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4MulPerElem_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

4-D vector in which each element is the product of the corresponding elements of the specified 4-D vectors

Description

Multiply two 4-D vectors element by element.

vmathSoaV4Neg_V

Negate all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Neg_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector containing negated elements of the specified 4-D vector

Description

Negate all elements of a 4-D vector.

vmathSoaV4Normalize_V

Normalize a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Normalize_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

The specified 4-D vector scaled to unit length

Description

Compute a normalized 4-D vector.

Notes

The result is unpredictable when all elements of vec are at or near zero.

vmathSoaV4Outer_V

Outer product of two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaV4Outer_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

The 4x4 matrix product of a column-vector, *vec0*, and a row-vector, *vec1*

Description

Compute the outer product of two 4-D vectors.

vmathSoaV4Print_V

Print a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4Print_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

None

Description

Print a 4-D vector. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV4Prints_V

Print a 4-D vector and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4Prints_V(
        VmathSoaVector4 vec,
        const char *name
);
```

Arguments

vec	4-D vector
name	String printed with the 4-D vector

Return Values

None

Description

Print a 4-D vector and an associated string identifier. Prints the 4-D vector transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaV4RecipPerElem_V

Compute the reciprocal of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4RecipPerElem_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector

Description

Create a 4-D vector in which each element is the reciprocal of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function recipf4.

vmathSoaV4RsqrtPerElem_V

Compute the reciprocal square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4RsqrtPerElem_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector

Description

Create a 4-D vector in which each element is the reciprocal square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathSoaV4ScalarDiv_V

Divide a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4ScalarDiv_V(
        VmathSoaVector4 vec,
        vec_float4 scalar
);
```

Arguments

vec	4-D vector
scalar	Scalar value

Return Values

Quotient of the specified 4-D vector and scalar

Description

Divide a 4-D vector by a scalar.

vmathSoaV4ScalarMul_V

Multiply a 4-D vector by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4ScalarMul_V(
        VmathSoaVector4 vec,
        vec_float4 scalar
);
```

Arguments

vec	4-D vector
scalar	Scalar value

Return Values

Product of the specified 4-D vector and scalar

Description

Multiply a 4-D vector by a scalar.

vmathSoaV4Select_V

Conditionally select between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Select_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1,
        vec_uint4 select1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector
select1	For each of the four word slots, this mask selects either the 4-D vector in the corresponding slot of *vec0* or the 4-D vector in the corresponding slot of *vec1*. A 0 bit selects from *vec0* whereas a 1 bit selects from *vec1*. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the 4-D vector at the corresponding slot of *vec0* or *vec1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *vec0*, and a value of 0xFFFFFFFF selects the slot of *vec1*

Description

Conditionally select one of the 4-D vectors at each of the corresponding slots of *vec0* or *vec1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaV4SetElem_V

Set an x, y, z, or w element of a 4-D vector by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4SetElem_V(
        VmathSoaVector4 *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output 4-D vector
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a 4-D vector by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaV4SetW_V

Set the w element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4SetW_V(
        VmathSoaVector4 *result,
        vec_float4 w
);
```

Arguments

result	An output 4-D vector
w	Scalar value

Return Values

None

Description

Set the w element of a 4-D vector to the specified scalar value.

vmathSoaV4SetX_V

Set the x element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4SetX_V(
        VmathSoaVector4 *result,
        vec_float4 x
);
```

Arguments

result	An output 4-D vector
x	Scalar value

Return Values

None

Description

Set the x element of a 4-D vector to the specified scalar value.

vmathSoaV4SetXYZ_V

Set the x, y, and z elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4SetXYZ_V(
        VmathSoaVector4 *result,
        VmathSoaVector3 vec
);
```

Arguments

result	An output 4-D vector
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathSoaV4SetY_V

Set the y element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4SetY_V(
        VmathSoaVector4 *result,
        vec_float4 y
);
```

Arguments

result	An output 4-D vector
y	Scalar value

Return Values

None

Description

Set the y element of a 4-D vector to the specified scalar value.

vmathSoaV4SetZ_V

Set the z element of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4SetZ_V(
        VmathSoaVector4 *result,
        vec_float4 z
);
```

Arguments

result	An output 4-D vector
z	Scalar value

Return Values

None

Description

Set the z element of a 4-D vector to the specified scalar value.

vmathSoaV4Slerp_V

Spherical linear interpolation between two 4-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Slerp_V(
        vec_float4 t,
        VmathSoaVector4 unitVec0,
        VmathSoaVector4 unitVec1
);
```

Arguments

t	Interpolation parameter
unitVec0	4-D vector, expected to be unit-length
unitVec1	4-D vector, expected to be unit-length

Return Values

Interpolated 4-D vector

Description

Perform spherical linear interpolation between two 4-D vectors.

Notes

The result is unpredictable if the vectors point in opposite directions. Does not clamp *t* between 0 and 1.

vmathSoaV4SqrtPerElem_V

Compute the square root of a 4-D vector per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4SqrtPerElem_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector

Description

Create a 4-D vector in which each element is the square root of the corresponding element of the specified 4-D vector.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathSoaV4StoreHalfFloats_V

Store four slots of an SoA 4-D vector as half-floats.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaV4StoreHalfFloats_V(
        VmathSoaVector4 vec,
        vec_ushort8 *twoQuads
);
```

Arguments

vec	4-D vector
twoQuads	An output array of 2 quadwords containing 16 half-floats

Return Values

None

Description

Store four slots of an SoA 4-D vector in two quadwords of half-float values. Numbering slots of *vec* as 0..3, the output is {x0,y0,z0,w0,x1,y1,z1,w1,x2,y2,z2,w2,x3,y3,z3,w3}.

vmathSoaV4Sub_V

Subtract a 4-D vector from another 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaV4Sub_V(
        VmathSoaVector4 vec0,
        VmathSoaVector4 vec1
);
```

Arguments

vec0	4-D vector
vec1	4-D vector

Return Values

Difference of the specified 4-D vectors

Description

Subtract a 4-D vector from another 4-D vector.

vmathSoaV4Sum_V

Compute the sum of all elements of a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaV4Sum_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

Sum of all elements of *vec*

Description

Compute the sum of all elements of a 4-D vector.

Point Functions (SoA, by value)

vmathSoaP3AbsPerElem_V

Compute the absolute value of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3AbsPerElem_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the absolute value of the corresponding element of pnt

Description

Compute the absolute value of each element of a 3-D point.

vmathSoaP3AddV3_V

Add a 3-D point to a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3AddV3_V(
        VmathSoaPoint3 pnt,
        VmathSoaVector3 vec
);
```

Arguments

pnt	3-D point
vec	3-D vector

Return Values

Sum of the specified 3-D point and 3-D vector

Description

Add a 3-D point to a 3-D vector.

vmathSoaP3CopySignPerElem_V

Copy sign from one 3-D point to another, per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3CopySignPerElem_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0 3-D point
pnt1 3-D point

Return Values

3-D point in which each element has the magnitude of the corresponding element of *pnt0* and the sign of the corresponding element of *pnt1*

Description

For each element, create a value composed of the magnitude of *pnt0* and the sign of *pnt1*.

vmathSoaP3Dist_V

Compute the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3Dist_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

Distance between two 3-D points

Description

Compute the distance between two 3-D points.

vmathSoaP3DistFromOrigin_V

Compute the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3DistFromOrigin_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Distance of a 3-D point from the origin

Description

Compute the distance of a 3-D point from the coordinate-system origin.

vmathSoaP3DistSqr_V

Compute the square of the distance between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3DistSqr_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0 3-D point
pnt1 3-D point

Return Values

Square of the distance between two 3-D points

Description

Compute the square of the distance between two 3-D points.

vmathSoaP3DistSqrFromOrigin_V

Compute the square of the distance of a 3-D point from the coordinate-system origin.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3DistSqrFromOrigin_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Square of the distance of a 3-D point from the origin

Description

Compute the square of the distance of a 3-D point from the coordinate-system origin.

vmathSoaP3DivPerElem_V

Divide two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3DivPerElem_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the quotient of the corresponding elements of the specified 3-D points

Description

Divide two 3-D points element by element.

Notes

Floating-point behavior matches standard library function divf4.

vmathSoaP3Get4Aos_V

Extract four AoS 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3Get4Aos_V(
        VmathSoaPoint3 pnt,
        VmathPoint3 *result0,
        VmathPoint3 *result1,
        VmathPoint3 *result2,
        VmathPoint3 *result3
);
```

Arguments

pnt	3-D point
result0	An output AoS 3-D point
result1	An output AoS 3-D point
result2	An output AoS 3-D point
result3	An output AoS 3-D point

Return Values

None

Description

Extract four AoS 3-D points from four slots of an SoA 3-D point (transpose the data format).

vmathSoaP3GetElem_V

Get an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3GetElem_V(
        VmathSoaPoint3 pnt,
        int idx
);
```

Arguments

pnt	3-D point
idx	Index, expected in the range 0-2

Return Values

Element selected by the specified index

Description

Get an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathSoaP3GetX_V

Get the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3GetX_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

x element of a 3-D point

Description

Get the x element of a 3-D point.

vmathSoaP3GetY_V

Get the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3GetY_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

y element of a 3-D point

Description

Get the y element of a 3-D point.

vmathSoaP3GetZ_V

Get the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3GetZ_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

z element of a 3-D point

Description

Get the z element of a 3-D point.

vmathSoaP3Lerp_V

Linear interpolation between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3Lerp_V(
        vec_float4 t,
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

t	Interpolation parameter
pnt0	3-D point
pnt1	3-D point

Return Values

Interpolated 3-D point

Description

Linearly interpolate between two 3-D points.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaP3LoadXYZArray_V

Load four three-float 3-D points, stored in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3LoadXYZArray_V(
        VmathSoaPoint3 *pnt,
        const vec_float4 *threeQuads
);
```

Arguments

pnt	An output 3-D point
threeQuads	Array of 3 quadwords containing 12 floats

Return Values

None

Description

Load four three-float 3-D points, stored in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}, into four slots of an SoA 3-D point.

vmathSoaP3MakeFrom4Aos_V

Insert four AoS 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MakeFrom4Aos_V(
        VmathPoint3 pnt0,
        VmathPoint3 pnt1,
        VmathPoint3 pnt2,
        VmathPoint3 pnt3
);
```

Arguments

pnt0	AoS 3-D point
pnt1	AoS 3-D point
pnt2	AoS 3-D point
pnt3	AoS 3-D point

Return Values

The constructed SoA 3-D point

Description

Insert four AoS 3-D points into four slots of an SoA 3-D point (transpose the data format).

vmathSoaP3MakeFromAos_V

Replicate an AoS 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MakeFromAos_V(
        VmathPoint3 pnt
);
```

Arguments

pnt AoS 3-D point

Return Values

The constructed SoA 3-D point

Description

Replicate an AoS 3-D point in all four slots of an SoA 3-D point.

vmathSoaP3MakeFromElems_V

Construct a 3-D point from x, y, and z elements.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MakeFromElems_V(
        vec_float4 x,
        vec_float4 y,
        vec_float4 z
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value

Return Values

The 3-D point that contains the specified elements

Description

Construct a 3-D point containing the specified x, y, and z elements.

vmathSoaP3MakeFromScalar_V

Set all elements of a 3-D point to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3-D point

Description

Construct a 3-D point with all elements set to the scalar value argument.

vmathSoaP3MakeFromV3_V

Copy elements from a 3-D vector into a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MakeFromV3_V(
        VmathSoaVector3 vec
);
```

Arguments

vec 3-D vector

Return Values

The constructed 3-D point

Description

Construct a 3-D point containing the x, y, and z elements of the specified 3-D vector.

vmathSoaP3MaxElem_V

Maximum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3MaxElem_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Maximum value of all elements of *pnt*

Description

Compute the maximum value of all elements of a 3-D point.

vmathSoaP3MaxPerElem_V

Maximum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MaxPerElem_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points

Description

Create a 3-D point in which each element is the maximum of the corresponding elements of the specified 3-D points.

vmathSoaP3MinElem_V

Minimum element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3MinElem_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Minimum value of all elements of *pnt*

Description

Compute the minimum value of all elements of a 3-D point.

vmathSoaP3MinPerElem_V

Minimum of two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MinPerElem_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point

Return Values

3-D point in which each element is the minimum of the corresponding elements of the specified 3-D points

Description

Create a 3-D point in which each element is the minimum of the corresponding elements of two specified 3-D points.

vmathSoaP3MulPerElem_V

Multiply two 3-D points per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3MulPerElem_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0 3-D point
pnt1 3-D point

Return Values

3-D point in which each element is the product of the corresponding elements of the specified 3-D points

Description

Multiply two 3-D points element by element.

vmathSoaP3NonUniformScale_V

Apply non-uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3NonUniformScale_V(
        VmathSoaPoint3 pnt,
        VmathSoaVector3 scaleVec
);
```

Arguments

pnt	3-D point
scaleVec	3-D vector

Return Values

3-D point in which each element is the product of the corresponding elements of the specified 3-D point and 3-D vector

Description

Apply non-uniform scale to a 3-D point.

vmathSoaP3Print_V

Print a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3Print_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

None

Description

Print a 3-D point. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaP3Prints_V

Print a 3-D point and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3Prints_V(
        VmathSoaPoint3 pnt,
        const char *name
);
```

Arguments

pnt	3-D point
name	String printed with the 3-D point

Return Values

None

Description

Print a 3-D point and an associated string identifier. Prints the 3-D point transposed, that is, as a row instead of a column.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaP3Projection_V

Scalar projection of a 3-D point on a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3Projection_V(
        VmathSoaPoint3 pnt,
        VmathSoaVector3 unitVec
);
```

Arguments

pnt	3-D point
unitVec	3-D vector, expected to be unit-length

Return Values

Scalar projection of the 3-D point on the unit-length 3-D vector

Description

Scalar projection of a 3-D point on a unit-length 3-D vector (dot product).

vmathSoaP3RecipPerElem_V

Compute the reciprocal of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3RecipPerElem_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point

Description

Create a 3-D point in which each element is the reciprocal of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function recipf4.

vmathSoaP3RsqrtPerElem_V

Compute the reciprocal square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3RsqrtPerElem_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point

Description

Create a 3-D point in which each element is the reciprocal square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function rsqrtf4.

vmathSoaP3Scale_V

Apply uniform scale to a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3Scale_V(
        VmathSoaPoint3 pnt,
        vec_float4 scaleVal
);
```

Arguments

pnt	3-D point
scaleVal	Scalar value

Return Values

3-D point in which every element is multiplied by the scalar value

Description

Apply uniform scale to a 3-D point.

vmathSoaP3Select_V

Conditionally select between two 3-D points.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3Select_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1,
        vec_uint4 select1
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
select1	For each of the four word slots, this mask selects either the 3-D point in the corresponding slot of *pnt0* or the 3-D point in the corresponding slot of *pnt1*. A 0 bit selects from *pnt0* whereas a 1 bit selects from *pnt1*. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the 3-D point at the corresponding slot of *pnt0* or *pnt1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *pnt0*, and a value of 0xFFFFFFFF selects the slot of *pnt1*

Description

Conditionally select one of the 3-D points at each of the corresponding slots of *pnt0* or *pnt1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaP3SetElem_V

Set an x, y, or z element of a 3-D point by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3SetElem_V(
        VmathSoaPoint3 *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output 3-D point
idx	Index, expected in the range 0-2
value	Scalar value

Return Values

None

Description

Set an x, y, or z element of a 3-D point by specifying an index of 0, 1, or 2, respectively.

vmathSoaP3SetX_V

Set the x element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3SetX_V(
        VmathSoaPoint3 *result,
        vec_float4 x
);
```

Arguments

result	An output 3-D point
x	Scalar value

Return Values

None

Description

Set the x element of a 3-D point to the specified scalar value.

vmathSoaP3SetY_V

Set the y element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3SetY_V(
        VmathSoaPoint3 *result,
        vec_float4 y
);
```

Arguments

result	An output 3-D point
y	Scalar value

Return Values

None

Description

Set the y element of a 3-D point to the specified scalar value.

vmathSoaP3SetZ_V

Set the z element of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3SetZ_V(
        VmathSoaPoint3 *result,
        vec_float4 z
);
```

Arguments

result	An output 3-D point
z	Scalar value

Return Values

None

Description

Set the z element of a 3-D point to the specified scalar value.

vmathSoaP3SqrtPerElem_V

Compute the square root of a 3-D point per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3SqrtPerElem_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

3-D point in which each element is the square root of the corresponding element of the specified 3-D point

Description

Create a 3-D point in which each element is the square root of the corresponding element of the specified 3-D point.

Notes

Floating-point behavior matches standard library function sqrtf4.

vmathSoaP3StoreHalfFloats_V

Store eight slots of two SoA 3-D points as half-floats.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3StoreHalfFloats_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1,
        vec_ushort8 *threeQuads
);
```

Arguments

pnt0	3-D point
pnt1	3-D point
threeQuads	An output array of 3 quadwords containing 24 half-floats

Return Values

None

Description

Store eight slots of two SoA 3-D points in three quadwords of half-float values. Numbering slots of *pnt0* as 0..3 and slots of *pnt1* as 4..7, the output is {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5,x6,y6,z6,x7,y7,z7}.

vmathSoaP3StoreXYZArray_V

Store four slots of an SoA 3-D point in three quadwords.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaP3StoreXYZArray_V(
        VmathSoaPoint3 pnt,
        vec_float4 *threeQuads
);
```

Arguments

pnt	3-D point
threeQuads	An output array of 3 quadwords containing 12 floats

Return Values

None

Description

Store four slots of an SoA 3-D point in three quadwords as {x0,y0,z0,x1,y1,z1,x2,y2,z2,x3,y3,z3}.

vmathSoaP3Sub_V

Subtract a 3-D point from another 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaP3Sub_V(
        VmathSoaPoint3 pnt0,
        VmathSoaPoint3 pnt1
);
```

Arguments

pnt0 3-D point
pnt1 3-D point

Return Values

Difference of the specified 3-D points

Description

Subtract a 3-D point from another 3-D point.

vmathSoaP3SubV3_V

Subtract a 3-D vector from a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaP3SubV3_V(
        VmathSoaPoint3 pnt,
        VmathSoaVector3 vec
);
```

Arguments

pnt	3-D point
vec	3-D vector

Return Values

Difference of the specified 3-D point and 3-D vector

Description

Subtract a 3-D vector from a 3-D point.

vmathSoaP3Sum_V

Compute the sum of all elements of a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaP3Sum_V(
        VmathSoaPoint3 pnt
);
```

Arguments

pnt 3-D point

Return Values

Sum of all elements of *pnt*

Description

Compute the sum of all elements of a 3-D point.

Quaternion Functions (SoA, by value)

vmathSoaQAdd_V

Add two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQAdd_V(
        VmathSoaQuat quat0,
        VmathSoaQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Sum of the specified quaternions

Description

Add two quaternions.

vmathSoaQConj_V

Compute the conjugate of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQConj_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

Conjugate of the specified quaternion

Description

Compute the conjugate of a quaternion.

vmathSoaQDot_V

Compute the dot product of two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQDot_V(
        VmathSoaQuat quat0,
        VmathSoaQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Dot product of the specified quaternions

Description

Compute the dot product of two quaternions.

vmathSoaQGet4Aos_V

Extract four AoS quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQGet4Aos_V(
        VmathSoaQuat quat,
        VmathQuat *result0,
        VmathQuat *result1,
        VmathQuat *result2,
        VmathQuat *result3
);
```

Arguments

quat	Quaternion
result0	An output AoS quaternion
result1	An output AoS quaternion
result2	An output AoS quaternion
result3	An output AoS quaternion

Return Values

None

Description

Extract four AoS quaternions from four slots of an SoA quaternion (transpose the data format).

vmathSoaQGetElem_V

Get an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQGetElem_V(
        VmathSoaQuat quat,
        int idx
);
```

Arguments

quat	Quaternion
idx	Index, expected in the range 0-3

Return Values

Element selected by the specified index

Description

Get an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaQGetW_V

Get the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQGetW_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

w element of a quaternion

Description

Get the w element of a quaternion.

vmathSoaQGetX_V

Get the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQGetX_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

x element of a quaternion

Description

Get the x element of a quaternion.

vmathSoaQGetXYZ_V

Get the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaQGetXYZ_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

3-D vector containing x, y, and z elements

Description

Extract a quaternion's x, y, and z elements into a 3-D vector.

vmathSoaQGetY_V

Get the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQGetY_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

y element of a quaternion

Description

Get the y element of a quaternion.

vmathSoaQGetZ_V

Get the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQGetZ_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

z element of a quaternion

Description

Get the z element of a quaternion.

vmathSoaQLength_V

Compute the length of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQLength_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

Length of the specified quaternion

Description

Compute the length of a quaternion.

vmathSoaQLerp_V

Linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQLerp_V(
        vec_float4 t,
        VmathSoaQuat quat0,
        VmathSoaQuat quat1
);
```

Arguments

t	Interpolation parameter
quat0	Quaternion
quat1	Quaternion

Return Values

Interpolated quaternion

Description

Linearly interpolate between two quaternions.

Notes

Does not clamp *t* between 0 and 1.

vmathSoaQMakeFrom4Aos_V

Insert four AoS quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFrom4Aos_V(
        VmathQuat quat0,
        VmathQuat quat1,
        VmathQuat quat2,
        VmathQuat quat3
);
```

Arguments

quat0	AoS quaternion
quat1	AoS quaternion
quat2	AoS quaternion
quat3	AoS quaternion

Return Values

The constructed SoA quaternion

Description

Insert four AoS quaternions into four slots of an SoA quaternion (transpose the data format).

vmathSoaQMakeFromAos_V

Replicate an AoS quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFromAos_V(
        VmathQuat quat
);
```

Arguments

quat AoS quaternion

Return Values

The constructed SoA quaternion

Description

Replicate an AoS quaternion in all four slots of an SoA quaternion.

vmathSoaQMakeFromElems_V

Construct a quaternion from x, y, z, and w elements.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFromElems_V(
        vec_float4 x,
        vec_float4 y,
        vec_float4 z,
        vec_float4 w
);
```

Arguments

x Scalar value
y Scalar value
z Scalar value
w Scalar value

Return Values

The quaternion that contains the specified elements

Description

Construct a quaternion containing the specified x, y, z, and w elements.

vmathSoaQMakeFromM3_V

Convert a rotation matrix to a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFromM3_V(
        VmathSoaMatrix3 rotMat
);
```

Arguments

rotMat 3x3 matrix, expected to be a rotation matrix

Return Values

The constructed quaternion

Description

Construct a unit-length quaternion representing the same transformation as a rotation matrix.

vmathSoaQMakeFromScalar_V

Set all elements of a quaternion to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion with all elements set to the scalar value argument.

vmathSoaQMakeFromV3Scalar_V

Construct a quaternion from a 3-D vector and a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFromV3Scalar_V(
        VmathSoaVector3 xyz,
        vec_float4 w
);
```

Arguments

xyz	3-D vector
w	Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion with the x, y, and z elements of the specified 3-D vector and with the w element set to the specified scalar.

vmathSoaQMakeFromV4_V

Copy elements from a 4-D vector into a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeFromV4_V(
        VmathSoaVector4 vec
);
```

Arguments

vec 4-D vector

Return Values

The constructed quaternion

Description

Construct a quaternion containing the x, y, z, and w elements of the specified 4-D vector.

vmathSoaQMakeIdentity_V

Construct an identity quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeIdentity_V();
```

Arguments

None

Return Values

The constructed quaternion

Description

Construct an identity quaternion equal to (0,0,0,1).

vmathSoaQMakeRotationArc_V

Construct a quaternion to rotate between two unit-length 3-D vectors.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeRotationArc_V(
        VmathSoaVector3 unitVec0,
        VmathSoaVector3 unitVec1
);
```

Arguments

unitVec0	3-D vector, expected to be unit-length
unitVec1	3-D vector, expected to be unit-length

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate between two unit-length 3-D vectors.

Notes

The result is unpredictable if *unitVec0* and *unitVec1* point in opposite directions.

vmathSoaQMakeRotationAxis_V

Construct a quaternion to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeRotationAxis_V(
        vec_float4 radians,
        VmathSoaVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaQMakeRotationX_V

Construct a quaternion to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeRotationX_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around the x axis by the specified radians angle.

vmathSoaQMakeRotationY_V

Construct a quaternion to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeRotationY_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around the y axis by the specified radians angle.

vmathSoaQMakeRotationZ_V

Construct a quaternion to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMakeRotationZ_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed quaternion

Description

Construct a quaternion to rotate around the z axis by the specified radians angle.

vmathSoaQMul_V

Multiply two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQMul_V(
        VmathSoaQuat quat0,
        VmathSoaQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Product of the specified quaternions

Description

Multiply two quaternions.

vmathSoaQNeg_V

Negate all elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQNeg_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

Quaternion containing negated elements of the specified quaternion

Description

Negate all elements of a quaternion.

vmathSoaQNorm_V

Compute the norm of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaQNorm_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

The norm of the specified quaternion

Description

Compute the norm, equal to the square of the length, of a quaternion.

vmathSoaQNormalize_V

Normalize a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQNormalize_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

The specified quaternion scaled to unit length

Description

Compute a normalized quaternion.

Notes

The result is unpredictable when all elements of quat are at or near zero.

vmathSoaQPrint_V

Print a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQPrint_V(
        VmathSoaQuat quat
);
```

Arguments

quat Quaternion

Return Values

None

Description

Print a quaternion.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaQPrints_V

Print a quaternion and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQPrints_V(
        VmathSoaQuat quat,
        const char *name
);
```

Arguments

quat	Quaternion
name	String printed with the quaternion

Return Values

None

Description

Print a quaternion and an associated string identifier.

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaQRotate_V

Use a unit-length quaternion to rotate a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaQRotate_V(
        VmathSoaQuat unitQuat,
        VmathSoaVector3 vec
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length
vec	3-D vector

Return Values

The rotated 3-D vector, equivalent to unitQuat*Quat(vec,0)*conj(unitQuat)

Description

Rotate a 3-D vector by applying a unit-length quaternion.

vmathSoaQScalarDiv_V

Divide a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQScalarDiv_V(
        VmathSoaQuat quat,
        vec_float4 scalar
);
```

Arguments

quat	Quaternion
scalar	Scalar value

Return Values

Quotient of the specified quaternion and scalar

Description

Divide a quaternion by a scalar.

vmathSoaQScalarMul_V

Multiply a quaternion by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQScalarMul_V(
        VmathSoaQuat quat,
        vec_float4 scalar
);
```

Arguments

quat	Quaternion
scalar	Scalar value

Return Values

Product of the specified quaternion and scalar

Description

Multiply a quaternion by a scalar.

vmathSoaQSelect_V

Conditionally select between two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQSelect_V(
        VmathSoaQuat quat0,
        VmathSoaQuat quat1,
        vec_uint4 select1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion
select1	For each of the four word slots, this mask selects either the quaternion in the corresponding slot of *quat0* or the quaternion in the corresponding slot of *quat1*. A 0 bit selects from *quat0* whereas a 1 bit selects from *quat1*. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the quaternion at the corresponding slot of *quat0* or *quat1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *quat0*, and a value of 0xFFFFFFFF selects the slot of *quat1*

Description

Conditionally select one of the quaternions at each of the corresponding slots of *quat0* or *quat1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaQSetElem_V

Set an x, y, z, or w element of a quaternion by index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQSetElem_V(
        VmathSoaQuat *result,
        int idx,
        vec_float4 value
);
```

Arguments

result	An output quaternion
idx	Index, expected in the range 0-3
value	Scalar value

Return Values

None

Description

Set an x, y, z, or w element of a quaternion by specifying an index of 0, 1, 2, or 3, respectively.

vmathSoaQSetW_V

Set the w element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQSetW_V(
        VmathSoaQuat *result,
        vec_float4 w
);
```

Arguments

result	An output quaternion
w	Scalar value

Return Values

None

Description

Set the w element of a quaternion to the specified scalar value.

vmathSoaQSetX_V

Set the x element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQSetX_V(
        VmathSoaQuat *result,
        vec_float4 x
);
```

Arguments

result	An output quaternion
x	Scalar value

Return Values

None

Description

Set the x element of a quaternion to the specified scalar value.

vmathSoaQSetXYZ_V

Set the x, y, and z elements of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQSetXYZ_V(
        VmathSoaQuat *result,
        VmathSoaVector3 vec
);
```

Arguments

result	An output quaternion
vec	3-D vector

Return Values

None

Description

Set the x, y, and z elements to those of the specified 3-D vector.

Notes

This function does not change the w element.

vmathSoaQSetY_V

Set the y element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQSetY_V(
        VmathSoaQuat *result,
        vec_float4 y
);
```

Arguments

result	An output quaternion
y	Scalar value

Return Values

None

Description

Set the y element of a quaternion to the specified scalar value.

vmathSoaQSetZ_V

Set the z element of a quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaQSetZ_V(
        VmathSoaQuat *result,
        vec_float4 z
);
```

Arguments

result	An output quaternion
z	Scalar value

Return Values

None

Description

Set the z element of a quaternion to the specified scalar value.

vmathSoaQSlerp_V

Spherical linear interpolation between two quaternions.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQSlerp_V(
        vec_float4 t,
        VmathSoaQuat unitQuat0,
        VmathSoaQuat unitQuat1
);
```

Arguments

t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length

Return Values

Interpolated quaternion

Description

Perform spherical linear interpolation between two quaternions.

Notes

Interpolates along the shortest path between orientations. Does not clamp *t* between 0 and 1.

vmathSoaQSquad_V

Spherical quadrangle interpolation.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQSquad_V(
        vec_float4 t,
        VmathSoaQuat unitQuat0,
        VmathSoaQuat unitQuat1,
        VmathSoaQuat unitQuat2,
        VmathSoaQuat unitQuat3
);
```

Arguments

t	Interpolation parameter
unitQuat0	Quaternion, expected to be unit-length
unitQuat1	Quaternion, expected to be unit-length
unitQuat2	Quaternion, expected to be unit-length
unitQuat3	Quaternion, expected to be unit-length

Return Values

Interpolated quaternion

Description

Perform spherical quadrangle interpolation between four quaternions.

vmathSoaQSub_V

Subtract a quaternion from another quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaQuat vmathSoaQSub_V(
        VmathSoaQuat quat0,
        VmathSoaQuat quat1
);
```

Arguments

quat0	Quaternion
quat1	Quaternion

Return Values

Difference of the specified quaternions

Description

Subtract a quaternion from another quaternion.

3x3 Matrix Functions (SoA, by value)

vmathSoaM3AbsPerElem_V

Compute the absolute value of a 3x3 matrix per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3AbsPerElem_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

3x3 matrix in which each element is the absolute value of the corresponding element of the specified 3x3 matrix

Description

Compute the absolute value of each element of a 3x3 matrix.

vmathSoaM3Add_V

Add two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Add_V(
        VmathSoaMatrix3 mat0,
        VmathSoaMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

Sum of the specified 3x3 matrices

Description

Add two 3x3 matrices.

vmathSoaM3AppendScale_V

Append (post-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3AppendScale_V(
        VmathSoaMatrix3 mat,
        VmathSoaVector3 scaleVec
);
```

Arguments

mat	3x3 matrix
scaleVec	3-D vector

Return Values

The product of *mat* and a scale transformation created from *scaleVec*

Description

Post-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM3Determinant_V

Determinant of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaM3Determinant_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 3x3 matrix.

vmathSoaM3Get4Aos_V

Extract four AoS 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3Get4Aos_V(
        VmathSoaMatrix3 mat,
        VmathMatrix3 *result0,
        VmathMatrix3 *result1,
        VmathMatrix3 *result2,
        VmathMatrix3 *result3
);
```

Arguments

mat	3x3 matrix
result0	An output AoS 3x3 matrix
result1	An output AoS 3x3 matrix
result2	An output AoS 3x3 matrix
result3	An output AoS 3x3 matrix

Return Values

None

Description

Extract four AoS 3x3 matrices from four slots of an SoA 3x3 matrix (transpose the data format).

vmathSoaM3GetCol0_V

Get column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM3GetCol0_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Column 0

Description

Get column 0 of a 3x3 matrix.

vmathSoaM3GetCol1_V

Get column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM3GetCol1_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Column 1

Description

Get column 1 of a 3x3 matrix.

vmathSoaM3GetCol2_V

Get column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM3GetCol2_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Column 2

Description

Get column 2 of a 3x3 matrix.

vmathSoaM3GetCol_V

Get the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM3GetCol_V(
        VmathSoaMatrix3 mat,
        int col
);
```

Arguments

mat	3x3 matrix
col	Index, expected in the range 0-2

Return Values

The column referred to by the specified index

Description

Get the column of a 3x3 matrix referred to by the specified index.

vmathSoaM3GetElem_V

Get the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaM3GetElem_V(
        VmathSoaMatrix3 mat,
        int col,
        int row
);
```

Arguments

mat	3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x3 matrix referred to by column and row indices.

vmathSoaM3GetRow_V

Get the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM3GetRow_V(
        VmathSoaMatrix3 mat,
        int row
);
```

Arguments

mat	3x3 matrix
row	Index, expected in the range 0-2

Return Values

The row referred to by the specified index

Description

Get the row of a 3x3 matrix referred to by the specified index.

vmathSoaM3Inverse_V

Compute the inverse of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Inverse_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

Inverse of *mat*

Description

Compute the inverse of a 3x3 matrix.

Notes

Result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathSoaM3MakeFrom4Aos_V

Insert four AoS 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeFrom4Aos_V(
        VmathMatrix3 mat0,
        VmathMatrix3 mat1,
        VmathMatrix3 mat2,
        VmathMatrix3 mat3
);
```

Arguments

mat0	AoS 3x3 matrix
mat1	AoS 3x3 matrix
mat2	AoS 3x3 matrix
mat3	AoS 3x3 matrix

Return Values

The constructed 3x3 matrix

Description

Insert four AoS 3x3 matrices into four slots of an SoA 3x3 matrix (transpose the data format).

vmathSoaM3MakeFromAos_V

Replicate an AoS 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeFromAos_V(
        VmathMatrix3 mat
);
```

Arguments

mat AoS 3x3 matrix

Return Values

The constructed 3x3 matrix

Description

Replicate an AoS 3x3 matrix in all four slots of an SoA 3x3 matrix.

vmathSoaM3MakeFromCols_V

Construct a 3x3 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeFromCols_V(
        VmathSoaVector3 col0,
        VmathSoaVector3 col1,
        VmathSoaVector3 col2
);
```

Arguments

col0	3-D vector
col1	3-D vector
col2	3-D vector

Return Values

The 3x3 matrix that contains the specified columns

Description

Construct a 3x3 matrix containing the specified columns.

vmathSoaM3MakeFromQ_V

Construct a 3x3 rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeFromQ_V(
        VmathSoaQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 3x3 matrix that applies the same rotation as *unitQuat*

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaM3MakeFromScalar_V

Set all elements of a 3x3 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix with all elements set to the scalar value argument.

vmathSoaM3MakeIdentity_V

Construct an identity 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeIdentity_V();
```

Arguments

None

Return Values

The constructed 3x3 matrix

Description

Construct an identity 3x3 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathSoaM3MakeRotationAxis_V

Construct a 3x3 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeRotationAxis_V(
        vec_float4 radians,
        VmathSoaVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaM3MakeRotationQ_V

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeRotationQ_V(
        VmathSoaQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 3x3 matrix that applies the same rotation as *unitQuat*

Description

Construct a 3x3 matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaM3MakeRotationX_V

Construct a 3x3 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeRotationX_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the x axis by the specified radians angle.

vmathSoaM3MakeRotationY_V

Construct a 3x3 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeRotationY_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the y axis by the specified radians angle.

vmathSoaM3MakeRotationZ_V

Construct a 3x3 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeRotationZ_V(
        vec_float4 radians
);
```

Arguments

radians	Scalar value

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the z axis by the specified radians angle.

vmathSoaM3MakeRotationZYX_V

Construct a 3x3 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeRotationZYX_V(
        VmathSoaVector3 radiansXYZ
);
```

Arguments

radiansXYZ 3-D vector

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to *rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())*.

vmathSoaM3MakeScale_V

Construct a 3x3 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MakeScale_V(
        VmathSoaVector3 scaleVec
);
```

Arguments

`scaleVec` 3-D vector

Return Values

The constructed 3x3 matrix

Description

Construct a 3x3 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *`scaleVec`*.

vmathSoaM3Mul_V

Multiply two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Mul_V(
        VmathSoaMatrix3 mat0,
        VmathSoaMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

Product of the specified 3x3 matrices

Description

Multiply two 3x3 matrices.

vmathSoaM3MulPerElem_V

Multiply two 3x3 matrices per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3MulPerElem_V(
        VmathSoaMatrix3 mat0,
        VmathSoaMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

3x3 matrix in which each element is the product of the corresponding elements of the specified 3x3 matrices

Description

Multiply two 3x3 matrices element by element.

vmathSoaM3MulV3_V

Multiply a 3x3 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM3MulV3_V(
        VmathSoaMatrix3 mat,
        VmathSoaVector3 vec
);
```

Arguments

mat	3x3 matrix
vec	3-D vector

Return Values

Product of the specified 3x3 matrix and 3-D vector

Description

Multiply a 3x3 matrix by a 3-D vector.

vmathSoaM3Neg_V

Negate all elements of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Neg_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

3x3 matrix containing negated elements of the specified 3x3 matrix

Description

Negate all elements of a 3x3 matrix.

vmathSoaM3PrependScale_V

Prepend (pre-multiply) a scale transformation to a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3PrependScale_V(
        VmathSoaVector3 scaleVec,
        VmathSoaMatrix3 mat
);
```

Arguments

scaleVec	3-D vector
mat	3x3 matrix

Return Values

The product of a scale transformation created from *scaleVec* and *mat*

Description

Pre-multiply a 3x3 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM3Print_V

Print a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3Print_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM3Prints_V

Print a 3x3 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3Prints_V(
        VmathSoaMatrix3 mat,
        const char *name
);
```

Arguments

mat	3x3 matrix
name	String printed with the 3x3 matrix

Return Values

None

Description

Print a 3x3 matrix and an associated string identifier. Unlike the printing of vectors, the 3x3 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM3ScalarMul_V

Multiply a 3x3 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3ScalarMul_V(
        VmathSoaMatrix3 mat,
        vec_float4 scalar
);
```

Arguments

mat	3x3 matrix
scalar	Scalar value

Return Values

Product of the specified 3x3 matrix and scalar

Description

Multiply a 3x3 matrix by a scalar.

vmathSoaM3Select_V

Conditionally select between two 3x3 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Select_V(
        VmathSoaMatrix3 mat0,
        VmathSoaMatrix3 mat1,
        vec_uint4 select1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix
select1	For each of the four word slots, this mask selects either the 3x3 matrix in the corresponding slot of `mat0` or the 3x3 matrix in the corresponding slot of `mat1`. A 0 bit selects from `mat0` whereas a 1 bit selects from `mat1`. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the 3x3 matrix at the corresponding slot of `mat0` or `mat1`, depending on the value of `select1` at the corresponding slot. A value of 0 selects the slot of `mat0` and a value of 0xFFFFFFFF selects the slot of `mat1`

Description

Conditionally select one of the 3x3 matrices at each of the corresponding slots of `mat0` or `mat1`.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaM3SetCol0_V

Set column 0 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3SetCol0_V(
        VmathSoaMatrix3 *result,
        VmathSoaVector3 col0
);
```

Arguments

result	An output 3x3 matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x3 matrix.

vmathSoaM3SetCol1_V

Set column 1 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3SetCol1_V(
        VmathSoaMatrix3 *result,
        VmathSoaVector3 col1
);
```

Arguments

result	An output 3x3 matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x3 matrix.

vmathSoaM3SetCol2_V

Set column 2 of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3SetCol2_V(
        VmathSoaMatrix3 *result,
        VmathSoaVector3 col2
);
```

Arguments

result	An output 3x3 matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x3 matrix.

vmathSoaM3SetCol_V

Set the column of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3SetCol_V(
        VmathSoaMatrix3 *result,
        int col,
        VmathSoaVector3 vec
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the column of a 3x3 matrix referred to by the specified index.

vmathSoaM3SetElem_V

Set the element of a 3x3 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3SetElem_V(
        VmathSoaMatrix3 *result,
        int col,
        int row,
        vec_float4 val
);
```

Arguments

result	An output 3x3 matrix
col	Index, expected in the range 0-2
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x3 matrix referred to by column and row indices.

vmathSoaM3SetRow_V

Set the row of a 3x3 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM3SetRow_V(
        VmathSoaMatrix3 *result,
        int row,
        VmathSoaVector3 vec
);
```

Arguments

result	An output 3x3 matrix
row	Index, expected in the range 0-2
vec	3-D vector

Return Values

None

Description

Set the row of a 3x3 matrix referred to by the specified index.

vmathSoaM3Sub_V

Subtract a 3x3 matrix from another 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Sub_V(
        VmathSoaMatrix3 mat0,
        VmathSoaMatrix3 mat1
);
```

Arguments

mat0	3x3 matrix
mat1	3x3 matrix

Return Values

Difference of the specified 3x3 matrices

Description

Subtract a 3x3 matrix from another 3x3 matrix.

vmathSoaM3Transpose_V

Transpose of a 3x3 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM3Transpose_V(
        VmathSoaMatrix3 mat
);
```

Arguments

mat 3x3 matrix

Return Values

mat transposed

Description

Compute the transpose of a 3x3 matrix.

4x4 Matrix Functions (SoA, by value)

vmathSoaM4AbsPerElem_V

Compute the absolute value of a 4x4 matrix per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4AbsPerElem_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

4x4 matrix in which each element is the absolute value of the corresponding element of the specified 4x4 matrix

Description

Compute the absolute value of each element of a 4x4 matrix.

vmathSoaM4Add_V

Add two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Add_V(
        VmathSoaMatrix4 mat0,
        VmathSoaMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

Sum of the specified 4x4 matrices

Description

Add two 4x4 matrices.

vmathSoaM4AffineInverse_V

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4AffineInverse_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Inverse of the specified 4x4 matrix

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is inverse(M), whose translation vector is -inverse(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions. The result is unpredictable when the determinant of *mat* is equal to or near 0.

vmathSoaM4AppendScale_V

Append (post-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4AppendScale_V(
        VmathSoaMatrix4 mat,
        VmathSoaVector3 scaleVec
);
```

Arguments

mat	4x4 matrix
scaleVec	3-D vector

Return Values

The product of `mat` and a scale transformation created from `scaleVec`

Description

Post-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM4Determinant_V

Determinant of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaM4Determinant_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

The determinant of *mat*

Description

Compute the determinant of a 4x4 matrix.

vmathSoaM4Get4Aos_V

Extract four AoS 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4Get4Aos_V(
        VmathSoaMatrix4 mat,
        VmathMatrix4 *result0,
        VmathMatrix4 *result1,
        VmathMatrix4 *result2,
        VmathMatrix4 *result3
);
```

Arguments

mat	4x4 matrix
result0	An output AoS 4x4 matrix
result1	An output AoS 4x4 matrix
result2	An output AoS 4x4 matrix
result3	An output AoS 4x4 matrix

Return Values

None

Description

Extract four AoS 4x4 matrices from four slots of an SoA 4x4 matrix (transpose the data format).

vmathSoaM4GetCol0_V

Get column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4GetCol0_V(
        VmathSoaMatrix4 mat
);
```

Arguments

`mat` 4x4 matrix

Return Values

Column 0

Description

Get column 0 of a 4x4 matrix.

vmathSoaM4GetCol1_V

Get column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4GetCol1_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Column 1

Description

Get column 1 of a 4x4 matrix.

vmathSoaM4GetCol2_V

Get column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4GetCol2_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Column 2

Description

Get column 2 of a 4x4 matrix.

vmathSoaM4GetCol3_V

Get column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4GetCol3_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Column 3

Description

Get column 3 of a 4x4 matrix.

vmathSoaM4GetCol_V

Get the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4GetCol_V(
        VmathSoaMatrix4 mat,
        int col
);
```

Arguments

mat	4x4 matrix
col	Index, expected in the range 0-3

Return Values

The column referred to by the specified index

Description

Get the column of a 4x4 matrix referred to by the specified index.

vmathSoaM4GetElem_V

Get the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaM4GetElem_V(
        VmathSoaMatrix4 mat,
        int col,
        int row
);
```

Arguments

mat	4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3

Return Values

Element selected by *col* and *row*

Description

Get the element of a 4x4 matrix referred to by column and row indices.

vmathSoaM4GetRow_V

Get the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4GetRow_V(
        VmathSoaMatrix4 mat,
        int row
);
```

Arguments

mat	4x4 matrix
row	Index, expected in the range 0-3

Return Values

The row referred to by the specified index

Description

Get the row of a 4x4 matrix referred to by the specified index.

vmathSoaM4GetTranslation_V

Get the translation component of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaM4GetTranslation_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Translation component

Description

Get the translation component of a 4x4 matrix.

vmathSoaM4GetUpper3x3_V

Get the upper-left 3x3 submatrix of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaM4GetUpper3x3_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Upper-left 3x3 submatrix

Description

Get the upper-left 3x3 submatrix of a 4x4 matrix.

vmathSoaM4Inverse_V

Compute the inverse of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Inverse_V(
        VmathSoaMatrix4 mat
);
```

Arguments

`mat` 4x4 matrix

Return Values

Inverse of `mat`

Description

Compute the inverse of a 4x4 matrix.

Notes

Result is unpredictable when the determinant of `mat` is equal to or near 0.

vmathSoaM4MakeFrom4Aos_V

Insert four AoS 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFrom4Aos_V(
        VmathMatrix4 mat0,
        VmathMatrix4 mat1,
        VmathMatrix4 mat2,
        VmathMatrix4 mat3
);
```

Arguments

mat0	AoS 4x4 matrix
mat1	AoS 4x4 matrix
mat2	AoS 4x4 matrix
mat3	AoS 4x4 matrix

Return Values

The constructed 4x4 matrix

Description

Insert four AoS 4x4 matrices into four slots of an SoA 4x4 matrix (transpose the data format).

vmathSoaM4MakeFromAos_V

Replicate an AoS 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFromAos_V(
        VmathMatrix4 mat
);
```

Arguments

mat AoS 4x4 matrix

Return Values

The constructed 4x4 matrix

Description

Replicate an AoS 4x4 matrix in all four slots of an SoA 4x4 matrix.

vmathSoaM4MakeFromCols_V

Construct a 4x4 matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFromCols_V(
        VmathSoaVector4 col0,
        VmathSoaVector4 col1,
        VmathSoaVector4 col2,
        VmathSoaVector4 col3
);
```

Arguments

col0	4-D vector
col1	4-D vector
col2	4-D vector
col3	4-D vector

Return Values

The 4x4 matrix that contains the specified columns

Description

Construct a 4x4 matrix containing the specified columns.

vmathSoaM4MakeFromM3V3_V

Construct a 4x4 matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFromM3V3_V(
        VmathSoaMatrix3 mat,
        VmathSoaVector3 translateVec
);
```

Arguments

mat	3x3 matrix
translateVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix whose upper 3x3 elements are equal to the 3x3 matrix argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathSoaM4MakeFromQV3_V

Construct a 4x4 matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFromQV3_V(
        VmathSoaQuat unitQuat,
        VmathSoaVector3 translateVec
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument, whose translation component is equal to the 3-D vector argument, and whose bottom row is (0,0,0,1).

vmathSoaM4MakeFromScalar_V

Set all elements of a 4x4 matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix with all elements set to the scalar value argument.

vmathSoaM4MakeFromT3_V

Construct a 4x4 matrix from a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFromT3_V(
        VmathSoaTransform3 mat
);
```

Arguments

mat 3x4 transformation matrix

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix whose upper 3x4 elements are equal to the 3x4 transformation matrix argument and whose bottom row is equal to (0,0,0,1).

vmathSoaM4MakeFrustum_V

Construct a perspective projection matrix based on frustum.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeFrustum_V(
        vec_float4 left,
        vec_float4 right,
        vec_float4 bottom,
        vec_float4 top,
        vec_float4 zNear,
        vec_float4 zFar
);
```

Arguments

left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a perspective projection matrix based on frustum, equal to:

```
2*zNear/(right-left)   0          (right+left)/(right-left)      0
       0       2*zNear/(top-bottom)  (top+bottom)/(top-bottom)      0
       0              0          -(zFar+zNear)/(zFar-zNear)
-2*zFar*zNear/(zFar-zNear)
       0              0                  -1                 0  .
```

vmathSoaM4MakeIdentity_V

Construct an identity 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeIdentity_V();
```

Arguments

None

Return Values

The constructed 4x4 matrix

Description

Construct an identity 4x4 matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathSoaM4MakeLookAt_V

Construct viewing matrix based on eye position, position looked at, and up direction.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeLookAt_V(
        VmathSoaPoint3 eyePos,
        VmathSoaPoint3 lookAtPos,
        VmathSoaVector3 upVec
);
```

Arguments

eyePos	3-D point
lookAtPos	3-D point
upVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct the inverse of a coordinate frame that is centered at the eye position, with z axis directed away from lookAtPos, and y axis oriented to best match the up direction.

vmathSoaM4MakeOrthographic_V

Construct an orthographic projection matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeOrthographic_V(
        vec_float4 left,
        vec_float4 right,
        vec_float4 bottom,
        vec_float4 top,
        vec_float4 zNear,
        vec_float4 zFar
);
```

Arguments

left	Scalar value
right	Scalar value
bottom	Scalar value
top	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct an orthographic projection matrix, equal to

```
2/(right-left)        0              0        -(right+left)/(right-left)
     0         2/(top-bottom)        0        -(top+bottom)/(top-bottom)
     0                0        -2/(zFar-zNear) -(zFar+zNear)/(zFar-zNear)
     0                0              0                   1  .
```

vmathSoaM4MakePerspective_V

Construct a perspective projection matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakePerspective_V(
        vec_float4 fovyRadians,
        vec_float4 aspect,
        vec_float4 zNear,
        vec_float4 zFar
);
```

Arguments

fovyRadians	Scalar value
aspect	Scalar value
zNear	Scalar value
zFar	Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a perspective projection matrix, equal to:

```
cot(fovyRadians/2)/aspect   0                   0                      0
       0             cot(fovyRadians/2)       0                      0
       0                      0   (zFar+zNear)/(zNear-zFar)
2*zFar*zNear/(zNear-zFar)
       0                      0             -1                     0  .
```

vmathSoaM4MakeRotationAxis_V

Construct a 4x4 matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeRotationAxis_V(
        vec_float4 radians,
        VmathSoaVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaM4MakeRotationQ_V

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeRotationQ_V(
        VmathSoaQuat unitQuat
);
```

Arguments

unitQuat Quaternion, expected to be unit-length

Return Values

A 4x4 matrix that applies the same rotation as *unitQuat*

Description

Construct a 4x4 matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaM4MakeRotationX_V

Construct a 4x4 matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeRotationX_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the x axis by the specified radians angle.

vmathSoaM4MakeRotationY_V

Construct a 4x4 matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeRotationY_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the y axis by the specified radians angle.

vmathSoaM4MakeRotationZ_V

Construct a 4x4 matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeRotationZ_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the z axis by the specified radians angle.

vmathSoaM4MakeRotationZYX_V

Construct a 4x4 matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeRotationZYX_V(
        VmathSoaVector3 radiansXYZ
);
```

Arguments

radiansXYZ 3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to *rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())*.

vmathSoaM4MakeScale_V

Construct a 4x4 matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeScale_V(
        VmathSoaVector3 scaleVec
);
```

Arguments

`scaleVec` 3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathSoaM4MakeTranslation_V

Construct a 4x4 matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MakeTranslation_V(
        VmathSoaVector3 translateVec
);
```

Arguments

translateVec	3-D vector

Return Values

The constructed 4x4 matrix

Description

Construct a 4x4 matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathSoaM4Mul_V

Multiply two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Mul_V(
        VmathSoaMatrix4 mat0,
        VmathSoaMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

Product of the specified 4x4 matrices

Description

Multiply two 4x4 matrices.

vmathSoaM4MulP3_V

Multiply a 4x4 matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4MulP3_V(
        VmathSoaMatrix4 mat,
        VmathSoaPoint3 pnt
);
```

Arguments

mat 4x4 matrix
pnt 3-D point

Return Values

Product of the specified 4x4 matrix and 3-D point

Description

Multiply a 4x4 matrix by a 3-D point treated as if it were a 4-D vector with the w element equal to 1.

vmathSoaM4MulPerElem_V

Multiply two 4x4 matrices per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MulPerElem_V(
        VmathSoaMatrix4 mat0,
        VmathSoaMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

4x4 matrix in which each element is the product of the corresponding elements of the specified 4x4 matrices

Description

Multiply two 4x4 matrices element by element.

vmathSoaM4MulT3_V

Multiply a 4x4 matrix by a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4MulT3_V(
        VmathSoaMatrix4 mat,
        VmathSoaTransform3 tfrm
);
```

Arguments

mat	4x4 matrix
tfrm	3x4 transformation matrix

Return Values

Product of the specified 4x4 matrix and 3x4 transformation matrix

Description

Multiply a 4x4 matrix by a 3x4 transformation matrix treated as if it were a 4x4 matrix with the bottom row equal to (0,0,0,1).

vmathSoaM4MulV3_V

Multiply a 4x4 matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4MulV3_V(
        VmathSoaMatrix4 mat,
        VmathSoaVector3 vec
);
```

Arguments

mat 4x4 matrix
vec 3-D vector

Return Values

Product of the specified 4x4 matrix and 3-D vector

Description

Multiply a 4x4 matrix by a 3-D vector treated as if it were a 4-D vector with the w element equal to 0.

vmathSoaM4MulV4_V

Multiply a 4x4 matrix by a 4-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaM4MulV4_V(
        VmathSoaMatrix4 mat,
        VmathSoaVector4 vec
);
```

Arguments

mat	4x4 matrix
vec	4-D vector

Return Values

Product of the specified 4x4 matrix and 4-D vector

Description

Multiply a 4x4 matrix by a 4-D vector.

vmathSoaM4Neg_V

Negate all elements of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Neg_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

4x4 matrix containing negated elements of the specified 4x4 matrix

Description

Negate all elements of a 4x4 matrix.

vmathSoaM4OrthoInverse_V

Compute the inverse of a 4x4 matrix, which is expected to be an affine matrix with an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4OrthoInverse_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

Inverse of the specified 4x4 matrix

Description

Naming the upper-left 3x3 submatrix of the specified 4x4 matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), whose translation vector is -transpose(M)*v, and whose bottom row is (0,0,0,1).

Notes

This can be used to achieve better performance than a general inverse when the specified 4x4 matrix meets the given restrictions.

vmathSoaM4PrependScale_V

Prepend (pre-multiply) a scale transformation to a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4PrependScale_V(
        VmathSoaVector3 scaleVec,
        VmathSoaMatrix4 mat
);
```

Arguments

scaleVec	3-D vector
mat	4x4 matrix

Return Values

The product of a scale transformation created from *scaleVec* and *mat*

Description

Pre-multiply a 4x4 matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaM4Print_V

Print a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4Print_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM4Prints_V

Print a 4x4 matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4Prints_V(
        VmathSoaMatrix4 mat,
        const char *name
);
```

Arguments

mat	4x4 matrix
name	String printed with the 4x4 matrix

Return Values

None

Description

Print a 4x4 matrix and an associated string identifier. Unlike the printing of vectors, the 4x4 matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaM4ScalarMul_V

Multiply a 4x4 matrix by a scalar.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4ScalarMul_V(
        VmathSoaMatrix4 mat,
        vec_float4 scalar
);
```

Arguments

mat	4x4 matrix
scalar	Scalar value

Return Values

Product of the specified 4x4 matrix and scalar

Description

Multiply a 4x4 matrix by a scalar.

vmathSoaM4Select_V

Conditionally select between two 4x4 matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Select_V(
        VmathSoaMatrix4 mat0,
        VmathSoaMatrix4 mat1,
        vec_uint4 select1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix
select1	For each of the four word slots, this mask selects either the 4x4 matrix in the corresponding slot of `mat0` or the 4x4 matrix in the corresponding slot of `mat1`. A 0 bit selects from `mat0` whereas a 1 bit selects from `mat1`. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the 4x4 matrix at the corresponding slot of `mat0` or `mat1`, depending on the value of `select1` at the corresponding slot. A value of 0 selects the slot of `mat0` and a value of 0xFFFFFFFF selects the slot of `mat1`

Description

Conditionally select one of the 4x4 matrices at each of the corresponding slots of `mat0` or `mat1`.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaM4SetCol0_V

Set column 0 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetCol0_V(
        VmathSoaMatrix4 *result,
        VmathSoaVector4 col0
);
```

Arguments

result	An output 4x4 matrix
col0	4-D vector

Return Values

None

Description

Set column 0 of a 4x4 matrix.

vmathSoaM4SetCol1_V

Set column 1 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetCol1_V(
        VmathSoaMatrix4 *result,
        VmathSoaVector4 col1
);
```

Arguments

result	An output 4x4 matrix
col1	4-D vector

Return Values

None

Description

Set column 1 of a 4x4 matrix.

vmathSoaM4SetCol2_V

Set column 2 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetCol2_V(
        VmathSoaMatrix4 *result,
        VmathSoaVector4 col2
);
```

Arguments

result	An output 4x4 matrix
col2	4-D vector

Return Values

None

Description

Set column 2 of a 4x4 matrix.

vmathSoaM4SetCol3_V

Set column 3 of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetCol3_V(
        VmathSoaMatrix4 *result,
        VmathSoaVector4 col3
);
```

Arguments

result	An output 4x4 matrix
col3	4-D vector

Return Values

None

Description

Set column 3 of a 4x4 matrix.

vmathSoaM4SetCol_V

Set the column of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetCol_V(
        VmathSoaMatrix4 *result,
        int col,
        VmathSoaVector4 vec
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the column of a 4x4 matrix referred to by the specified index.

vmathSoaM4SetElem_V

Set the element of a 4x4 matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetElem_V(
        VmathSoaMatrix4 *result,
        int col,
        int row,
        vec_float4 val
);
```

Arguments

result	An output 4x4 matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-3
val	Scalar value

Return Values

None

Description

Set the element of a 4x4 matrix referred to by column and row indices.

vmathSoaM4SetRow_V

Set the row of a 4x4 matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetRow_V(
        VmathSoaMatrix4 *result,
        int row,
        VmathSoaVector4 vec
);
```

Arguments

result	An output 4x4 matrix
row	Index, expected in the range 0-3
vec	4-D vector

Return Values

None

Description

Set the row of a 4x4 matrix referred to by the specified index.

vmathSoaM4SetTranslation_V

Set translation component.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetTranslation_V(
        VmathSoaMatrix4 *result,
        VmathSoaVector3 translateVec
);
```

Arguments

result	An output 4x4 matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 4x4 matrix equal to the specified 3-D vector.

Notes

This function does not change the bottom row elements.

vmathSoaM4SetUpper3x3_V

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaM4SetUpper3x3_V(
        VmathSoaMatrix4 *result,
        VmathSoaMatrix3 mat3
);
```

Arguments

result	An output 4x4 matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 4x4 matrix equal to the specified 3x3 matrix.

Notes

This function does not change the bottom row elements.

vmathSoaM4Sub_V

Subtract a 4x4 matrix from another 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Sub_V(
        VmathSoaMatrix4 mat0,
        VmathSoaMatrix4 mat1
);
```

Arguments

mat0	4x4 matrix
mat1	4x4 matrix

Return Values

Difference of the specified 4x4 matrices

Description

Subtract a 4x4 matrix from another 4x4 matrix.

vmathSoaM4Transpose_V

Transpose of a 4x4 matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix4 vmathSoaM4Transpose_V(
        VmathSoaMatrix4 mat
);
```

Arguments

mat 4x4 matrix

Return Values

mat transposed

Description

Compute the transpose of a 4x4 matrix.

Transformation Functions (SoA, by value)

vmathSoaT3AbsPerElem_V

Compute the absolute value of a 3x4 transformation matrix per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3AbsPerElem_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

3x4 transformation matrix in which each element is the absolute value of the corresponding element of the specified 3x4 transformation matrix

Description

Compute the absolute value of each element of a 3x4 transformation matrix.

vmathSoaT3AppendScale_V

Append (post-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3AppendScale_V(
        VmathSoaTransform3 tfrm,
        VmathSoaVector3 scaleVec
);
```

Arguments

tfrm	3x4 transformation matrix
scaleVec	3-D vector

Return Values

The product of *tfrm* and a scale transformation created from *scaleVec*

Description

Post-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaT3Get4Aos_V

Extract four AoS 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3Get4Aos_V(
        VmathSoaTransform3 tfrm,
        VmathTransform3 *result0,
        VmathTransform3 *result1,
        VmathTransform3 *result2,
        VmathTransform3 *result3
);
```

Arguments

tfrm	3x4 transformation matrix
result0	An output AoS 3x4 transformation matrix
result1	An output AoS 3x4 transformation matrix
result2	An output AoS 3x4 transformation matrix
result3	An output AoS 3x4 transformation matrix

Return Values

None

Description

Extract four AoS 3x4 transformation matrices from four slots of an SoA 3x4 transformation matrix (transpose the data format).

vmathSoaT3GetCol0_V

Get column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3GetCol0_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 0

Description

Get column 0 of a 3x4 transformation matrix.

vmathSoaT3GetCol1_V

Get column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3GetCol1_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 1

Description

Get column 1 of a 3x4 transformation matrix.

vmathSoaT3GetCol2_V

Get column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3GetCol2_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 2

Description

Get column 2 of a 3x4 transformation matrix.

vmathSoaT3GetCol3_V

Get column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3GetCol3_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Column 3

Description

Get column 3 of a 3x4 transformation matrix.

vmathSoaT3GetCol_V

Get the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3GetCol_V(
        VmathSoaTransform3 tfrm,
        int col
);
```

Arguments

tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3

Return Values

The column referred to by the specified index

Description

Get the column of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3GetElem_V

Get the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline vec_float4 vmathSoaT3GetElem_V(
        VmathSoaTransform3 tfrm,
        int col,
        int row
);
```

Arguments

tfrm	3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2

Return Values

Element selected by *col* and *row*

Description

Get the element of a 3x4 transformation matrix referred to by column and row indices.

vmathSoaT3GetRow_V

Get the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector4 vmathSoaT3GetRow_V(
        VmathSoaTransform3 tfrm,
        int row
);
```

Arguments

tfrm	3x4 transformation matrix
row	Index, expected in the range 0-2

Return Values

The row referred to by the specified index

Description

Get the row of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3GetTranslation_V

Get the translation component of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3GetTranslation_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Translation component

Description

Get the translation component of a 3x4 transformation matrix.

vmathSoaT3GetUpper3x3_V

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaMatrix3 vmathSoaT3GetUpper3x3_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Upper-left 3x3 submatrix

Description

Get the upper-left 3x3 submatrix of a 3x4 transformation matrix.

vmathSoaT3Inverse_V

Inverse of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3Inverse_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Inverse of *tfrm*

Description

Compute the inverse of a 3x4 transformation matrix.

Notes

Result is unpredictable when the determinant of the left 3x3 submatrix is equal to or near 0.

vmathSoaT3MakeFrom4Aos_V

Insert four AoS 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeFrom4Aos_V(
        VmathTransform3 tfrm0,
        VmathTransform3 tfrm1,
        VmathTransform3 tfrm2,
        VmathTransform3 tfrm3
);
```

Arguments

tfrm0	AoS 3x4 transformation matrix
tfrm1	AoS 3x4 transformation matrix
tfrm2	AoS 3x4 transformation matrix
tfrm3	AoS 3x4 transformation matrix

Return Values

The constructed 3x4 transformation matrix

Description

Insert four AoS 3x4 transformation matrices into four slots of an SoA 3x4 transformation matrix (transpose the data format).

vmathSoaT3MakeFromAos_V

Replicate an AoS 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeFromAos_V(
        VmathTransform3 tfrm
);
```

Arguments

tfrm AoS 3x4 transformation matrix

Return Values

The constructed 3x4 transformation matrix

Description

Replicate an AoS 3x4 transformation matrix in all four slots of an SoA 3x4 transformation matrix.

vmathSoaT3MakeFromCols_V

Construct a 3x4 transformation matrix containing the specified columns.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeFromCols_V(
        VmathSoaVector3 col0,
        VmathSoaVector3 col1,
        VmathSoaVector3 col2,
        VmathSoaVector3 col3
);
```

Arguments

col0	3-D vector
col1	3-D vector
col2	3-D vector
col3	3-D vector

Return Values

The 3x4 transformation matrix that contains the specified columns

Description

Construct a 3x4 transformation matrix containing the specified columns.

vmathSoaT3MakeFromM3V3_V

Construct a 3x4 transformation matrix from a 3x3 matrix and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeFromM3V3_V(
        VmathSoaMatrix3 tfrm,
        VmathSoaVector3 translateVec
);
```

Arguments

tfrm	3x3 matrix
translateVec	3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix whose upper 3x3 elements are equal to the 3x3 matrix argument and whose translation component is equal to the 3-D vector argument.

vmathSoaT3MakeFromQV3_V

Construct a 3x4 transformation matrix from a unit-length quaternion and a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeFromQV3_V(
        VmathSoaQuat unitQuat,
        VmathSoaVector3 translateVec
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length
translateVec	3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix whose upper-left 3x3 submatrix is a rotation matrix converted from the unit-length quaternion argument and whose translation component is equal to the 3-D vector argument.

vmathSoaT3MakeFromScalar_V

Set all elements of a 3x4 transformation matrix to the same scalar value.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeFromScalar_V(
        vec_float4 scalar
);
```

Arguments

scalar Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix with all elements set to the scalar value argument.

vmathSoaT3MakeIdentity_V

Construct an identity 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeIdentity_V();
```

Arguments

None

Return Values

The constructed 3x4 transformation matrix

Description

Construct an identity 3x4 transformation matrix in which non-diagonal elements are zero and diagonal elements are 1.

vmathSoaT3MakeRotationAxis_V

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeRotationAxis_V(
        vec_float4 radians,
        VmathSoaVector3 unitVec
);
```

Arguments

radians	Scalar value
unitVec	3-D vector, expected to be unit-length

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around a unit-length 3-D vector by the specified radians angle.

vmathSoaT3MakeRotationQ_V

Construct a rotation matrix from a unit-length quaternion.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeRotationQ_V(
        VmathSoaQuat unitQuat
);
```

Arguments

unitQuat	Quaternion, expected to be unit-length

Return Values

A 3x4 transformation matrix that applies the same rotation as *unitQuat*

Description

Construct a 3x4 transformation matrix that applies the same rotation as the specified unit-length quaternion.

vmathSoaT3MakeRotationX_V

Construct a 3x4 transformation matrix to rotate around the x axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeRotationX_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the x axis by the specified radians angle.

vmathSoaT3MakeRotationY_V

Construct a 3x4 transformation matrix to rotate around the y axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeRotationY_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the y axis by the specified radians angle.

vmathSoaT3MakeRotationZ_V

Construct a 3x4 transformation matrix to rotate around the z axis.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeRotationZ_V(
        vec_float4 radians
);
```

Arguments

radians Scalar value

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the z axis by the specified radians angle.

vmathSoaT3MakeRotationZYX_V

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeRotationZYX_V(
        VmathSoaVector3 radiansXYZ
);
```

Arguments

radiansXYZ 3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to rotate around the x, y, and z axes by the radians angles contained in a 3-D vector. Equivalent to *rotationZ(radiansXYZ.getZ()) * rotationY(radiansXYZ.getY()) * rotationX(radiansXYZ.getX())*.

vmathSoaT3MakeScale_V

Construct a 3x4 transformation matrix to perform scaling.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeScale_V(
        VmathSoaVector3 scaleVec
);
```

Arguments

scaleVec 3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to perform scaling, in which the non-diagonal elements are zero and the diagonal elements are set to the elements of *scaleVec*.

vmathSoaT3MakeTranslation_V

Construct a 3x4 transformation matrix to perform translation.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MakeTranslation_V(
        VmathSoaVector3 translateVec
);
```

Arguments

translateVec 3-D vector

Return Values

The constructed 3x4 transformation matrix

Description

Construct a 3x4 transformation matrix to perform translation, which is an identity matrix except for the translation component, with coordinates equal to those in *translateVec*.

vmathSoaT3Mul_V

Multiply two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3Mul_V(
        VmathSoaTransform3 tfrm0,
        VmathSoaTransform3 tfrm1
);
```

Arguments

tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix

Return Values

Product of the specified 3x4 transformation matrices

Description

Multiply two 3x4 transformation matrices.

vmathSoaT3MulP3_V

Multiply a 3x4 transformation matrix by a 3-D point.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaPoint3 vmathSoaT3MulP3_V(
        VmathSoaTransform3 tfrm,
        VmathSoaPoint3 pnt
);
```

Arguments

tfrm	3x4 transformation matrix
pnt	3-D point

Return Values

Product of the specified 3x4 transformation matrix and 3-D point

Description

Applies the 3x3 upper-left submatrix and the translation component of a 3x4 transformation matrix to a 3-D point.

vmathSoaT3MulPerElem_V

Multiply two 3x4 transformation matrices per element.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3MulPerElem_V(
        VmathSoaTransform3 tfrm0,
        VmathSoaTransform3 tfrm1
);
```

Arguments

tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix

Return Values

3x4 transformation matrix in which each element is the product of the corresponding elements of the specified 3x4 transformation matrices

Description

Multiply two 3x4 transformation matrices element by element.

vmathSoaT3MulV3_V

Multiply a 3x4 transformation matrix by a 3-D vector.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaVector3 vmathSoaT3MulV3_V(
        VmathSoaTransform3 tfrm,
        VmathSoaVector3 vec
);
```

Arguments

tfrm	3x4 transformation matrix
vec	3-D vector

Return Values

Product of the specified 3x4 transformation matrix and 3-D vector

Description

Applies the 3x3 upper-left submatrix (but not the translation component) of a 3x4 transformation matrix to a 3-D vector.

vmathSoaT3OrthoInverse_V

Compute the inverse of a 3x4 transformation matrix, expected to have an orthogonal upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3OrthoInverse_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

Inverse of the specified 3x4 transformation matrix

Description

Naming the upper-left 3x3 submatrix of the specified 3x4 transformation matrix as M, and its translation component as v, compute a matrix whose upper-left 3x3 submatrix is transpose(M), and whose translation vector is -transpose(M)*v.

Notes

This can be used to achieve better performance than a general inverse when the specified 3x4 transformation matrix meets the given restrictions.

vmathSoaT3PrependScale_V

Prepend (pre-multiply) a scale transformation to a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3PrependScale_V(
        VmathSoaVector3 scaleVec,
        VmathSoaTransform3 tfrm
);
```

Arguments

scaleVec	3-D vector
tfrm	3x4 transformation matrix

Return Values

The product of a scale transformation created from *scaleVec* and *tfrm*

Description

Pre-multiply a 3x4 transformation matrix by a scale transformation whose diagonal scale factors are contained in the 3-D vector.

Notes

Faster than creating and multiplying a scale transformation matrix.

vmathSoaT3Print_V

Print a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3Print_V(
        VmathSoaTransform3 tfrm
);
```

Arguments

tfrm 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaT3Prints_V

Print a 3x4 transformation matrix and an associated string identifier.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3Prints_V(
        VmathSoaTransform3 tfrm,
        const char *name
);
```

Arguments

tfrm	3x4 transformation matrix
name	String printed with the 3x4 transformation matrix

Return Values

None

Description

Print a 3x4 transformation matrix and an associated string identifier. Unlike the printing of vectors, the 3x4 transformation matrix is printed with the correct orientation (columns appear vertically).

Notes

Function is only defined when _VECTORMATH_DEBUG is defined.

vmathSoaT3Select_V

Conditionally select between two 3x4 transformation matrices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline VmathSoaTransform3 vmathSoaT3Select_V(
        VmathSoaTransform3 tfrm0,
        VmathSoaTransform3 tfrm1,
        vec_uint4 select1
);
```

Arguments

tfrm0	3x4 transformation matrix
tfrm1	3x4 transformation matrix
select1	For each of the four word slots, this mask selects either the 3x4 transformation matrix in the corresponding slot of *tfrm0* or the 3x4 transformation matrix in the corresponding slot of *tfrm1*. A 0 bit selects from *tfrm0* whereas a 1 bit selects from *tfrm1*. Identical bits should be set for each word of the mask.

Return Values

Each slot of the result is equal to the 3x4 transformation matrix at the corresponding slot of *tfrm0* or *tfrm1*, depending on the value of *select1* at the corresponding slot. A value of 0 selects the slot of *tfrm0* and a value of 0xFFFFFFFF selects the slot of *tfrm1*

Description

Conditionally select one of the 3x4 transformation matrices at each of the corresponding slots of *tfrm0* or *tfrm1*.

Notes

This function uses a conditional select instruction to avoid a branch.

vmathSoaT3SetCol0_V

Set column 0 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetCol0_V(
        VmathSoaTransform3 *result,
        VmathSoaVector3 col0
);
```

Arguments

result	An output 3x4 transformation matrix
col0	3-D vector

Return Values

None

Description

Set column 0 of a 3x4 transformation matrix.

vmathSoaT3SetCol1_V

Set column 1 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetCol1_V(
        VmathSoaTransform3 *result,
        VmathSoaVector3 col1
);
```

Arguments

result	An output 3x4 transformation matrix
col1	3-D vector

Return Values

None

Description

Set column 1 of a 3x4 transformation matrix.

vmathSoaT3SetCol2_V

Set column 2 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetCol2_V(
        VmathSoaTransform3 *result,
        VmathSoaVector3 col2
);
```

Arguments

result	An output 3x4 transformation matrix
col2	3-D vector

Return Values

None

Description

Set column 2 of a 3x4 transformation matrix.

vmathSoaT3SetCol3_V

Set column 3 of a 3x4 transformation matrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetCol3_V(
        VmathSoaTransform3 *result,
        VmathSoaVector3 col3
);
```

Arguments

result	An output 3x4 transformation matrix
col3	3-D vector

Return Values

None

Description

Set column 3 of a 3x4 transformation matrix.

vmathSoaT3SetCol_V

Set the column of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetCol_V(
        VmathSoaTransform3 *result,
        int col,
        VmathSoaVector3 vec
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
vec	3-D vector

Return Values

None

Description

Set the column of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3SetElem_V

Set the element of a 3x4 transformation matrix referred to by column and row indices.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetElem_V(
        VmathSoaTransform3 *result,
        int col,
        int row,
        vec_float4 val
);
```

Arguments

result	An output 3x4 transformation matrix
col	Index, expected in the range 0-3
row	Index, expected in the range 0-2
val	Scalar value

Return Values

None

Description

Set the element of a 3x4 transformation matrix referred to by column and row indices.

vmathSoaT3SetRow_V

Set the row of a 3x4 transformation matrix referred to by the specified index.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetRow_V(
        VmathSoaTransform3 *result,
        int row,
        VmathSoaVector4 vec
);
```

Arguments

result	An output 3x4 transformation matrix
row	Index, expected in the range 0-2
vec	4-D vector

Return Values

None

Description

Set the row of a 3x4 transformation matrix referred to by the specified index.

vmathSoaT3SetTranslation_V

Set translation component.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetTranslation_V(
        VmathSoaTransform3 *result,
        VmathSoaVector3 translateVec
);
```

Arguments

result	An output 3x4 transformation matrix
translateVec	3-D vector

Return Values

None

Description

Set the translation component of a 3x4 transformation matrix equal to the specified 3-D vector.

vmathSoaT3SetUpper3x3_V

Set the upper-left 3x3 submatrix.

Definition

```
#include <vectormath/c/vectormath_soa_v.h>
static inline void vmathSoaT3SetUpper3x3_V(
        VmathSoaTransform3 *result,
        VmathSoaMatrix3 mat3
);
```

Arguments

result	An output 3x4 transformation matrix
mat3	3x3 matrix

Return Values

None

Description

Set the upper-left 3x3 submatrix elements of a 3x4 transformation matrix equal to the specified 3x3 matrix.