vDSP Vector Scalar Arithmetic Operations Reference

Performance > Carbon

2007-06-15

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

vDSP Vector Scalar Arithmetic Operations Reference

Framework: Accelerate/vecLib

Declared in vDSP.h

Overview

Describes the C API for the vecLib functions that perform arithmetic operations combining a scalar with each element of a vector.

Functions by Task

Adding a Scalar to Elements of a Vector

`vDSP_vsadd` (page 8)
Vector scalar add; single precision.

`vDSP_vsaddD` (page 9)
Vector scalar add; double precision.

`vDSP_vsaddi` (page 10)
Integer vector scalar add.

Dividing Elements of a Vector by a Scalar

`vDSP_vsdiv` (page 10)
Vector scalar divide; single precision.

`vDSP_vsdivD` (page 11)
Vector scalar divide; double precision.

`vDSP_vsdivi` (page 12)
Integer vector scalar divide.

`vDSP_zvdiv` (page 19)
Complex vector divide; single precision.

`vDSP_zvdivD` (page 20)
Complex vector divide; double precision.

`vDSP_svdiv` (page 6)
Divide scalar by vector; single precision.

Multiplying Elements of a Vector by a Scalar

Multiplying Elements of a Vector by a Scalar, then Adding or Subtracting Another Scalar

Functions

vDSP_svdiv

Divide scalar by vector; single precision.

```
void vDSP_svdiv (float * A,
float * B,
vDSP_Stride J,
float * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Single-precision real input scalar

B

Single-precision real input vector

J

Stride for `B`

C

Single-precision real output vector

K

Stride for `C`

N

Count

Discussion
This performs the operation

$$C_{nK} = \frac{A}{B_{nJ}}, \qquad \text{n} = \{0, \text{N-1}\}$$

Divides scalar `A` by each element of vector `B`, storing the results in vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_svdivD

Divide scalar by vector; double precision.

```
void vDSP_svdivD (double * A,
double * B,
vDSP_Stride J,
double * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Double-precision real input scalar

B

Double-precision real input vector

J

Stride for `B`

C

Double-precision real output vector

K

Stride for `C`

N

Count

Discussion
This performs the operation

$$C_{nK} = \frac{A}{B_{nJ}}, \qquad n = \{0, N\text{-}1\}$$

Divides scalar `A` by each element of vector `B`, storing the results in vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsadd

Vector scalar add; single precision.

```
void vDSP_vsadd (float * A,
vDSP_Stride I,
float * B,
float * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Single-precision real input vector

I

Stride for `A`

B

Single-precision real input scalar

C

Single-precision real output vector

K

Stride for `C`

N

Count

Discussion
Performs the operation

$$C_{nK} = A_{nI} + B \qquad n = \{0, N\text{-}1\}$$

Adds scalar `B` to each element of vector `A` and stores the result in the corresponding element of vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsaddD

Vector scalar add; double precision.

```
void vDSP_vsaddD (double * A,
vDSP_Stride I,
double * B,
double * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A
: Double-precision real input vector

I
: Stride for `A`

B
: Double-precision real input scalar

C
: Double-precision real output vector

K
: Stride for `C`

N
: Count

Discussion
Performs the operation

$$C_{nK} = A_{nI} + B \qquad n = \{0, N\text{-}1\}$$

Adds scalar `B` to each element of vector `A` and stores the result in the corresponding element of vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsaddi

Integer vector scalar add.

```
void vDSP_vsaddi (int * A,
vDSP_Stride I,
int * B,
int * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Integer input vector

I

Stride for `A`

B

Integer input scalar

C

Integer output vector

K

Stride for `C`

N

Count

Discussion
Performs the operation

$$C_{nK} = A_{nI} + B \qquad n = \{0, N\text{-}1\}$$

Adds scalar `B` to each element of vector `A` and stores the result in the corresponding element of vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsdiv

Vector scalar divide; single precision.

```
void vDSP_vsdiv (float * A,
vDSP_Stride I,
float * B,
float * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Single-precision real input vector

I

Stride for `A`

B

Single-precision real input scalar

C

Single-precision real output vector

K

Stride for `C`

N

Count

Discussion
Performs the operation

$$C_{nK} = \frac{A_{nI}}{B} \qquad \text{n} = \{0, \text{N-1}\}$$

Divides each element of vector `A` by scalar `B` and stores the result in the corresponding element of vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsdivD

Vector scalar divide; double precision.

```
void vDSP_vsdivD (double * A,
vDSP_Stride I,
double * B,
double * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Double-precision real input vector

I

Stride for `A`

B

Double-precision real input scalar

C

Double-precision real output vector

K

Stride for `C`

N

Count

Discussion
Performs the operation

$$C_{nK} = \frac{A_{nI}}{B} \qquad n = \{0, N\text{-}1\}$$

Divides each element of vector `A` by scalar `B` and stores the result in the corresponding element of vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsdivi

Integer vector scalar divide.

```
void vDSP_vsdivi (int * A,
vDSP_Stride I,
int * B,
int * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A
: Integer input vector

I
: Stride for `A`

B
: Integer input scalar

C
: Integer output vector

K
: Stride for `C`

N
: Count

Discussion
Performs the operation

$$C_{nK} = \frac{A_{nI}}{B} \qquad n = \{0, N\text{-}1\}$$

Divides each element of vector `A` by scalar `B` and stores the result in the corresponding element of vector `C`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
vDSP.h

vDSP_vsma

Vector scalar multiply and vector add; single precision.

```
void vDSP_vsma (const float * A,
vDSP_Stride I,
const float * B,
const float * C,
vDSP_Stride K,
float * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A
: Single-precision real input vector

I
: Stride for A

B
: Single-precision real input scalar

C
: Single-precision real input vector

K
: Stride for C

D
: Single-precision real output vector

L
: Stride for D

N
: Count

Discussion
Performs the operation

$$D_{nM} = A_{nI} \bullet B + C_{nK} \qquad n = \{0, N\text{-}1\}$$

Multiplies vector A by scalar B and then adds the products to vector C. Results are stored in vector D.

Availability
Available in Mac OS X v10.4 and later.

Declared In
vDSP.h

vDSP_vsmaD

Vector scalar multiply and vector add; double precision.

```
void vDSP_vsmaD (const double * A,
vDSP_Stride I,
const double * B,
const double * C,
vDSP_Stride K,
double * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Double-precision real input vector

I

Stride for `A`

B

Double-precision real input scalar

C

Double-precision real input vector

K

Stride for `C`

D

Double-precision real output vector

L

Stride for `D`

N

Count

Discussion
Performs the operation

$$D_{nM} = A_{nI} \cdot B + C_{nK} \qquad n = \{0, N\text{-}1\}$$

Multiplies vector `A` by scalar `B` and then adds the products to vector `C`. Results are stored in vector `D`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsmsa

Vector scalar multiply and scalar add; single precision.

```
void vDSP_vsmsa (float * A,
vDSP_Stride I,
float * B,
float * C,
float * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Single-precision real input vector

I

Stride for `A`

B

Single-precision real input scalar

C

Single-precision real input scalar

D

Single-precision real output vector

L

Stride for `D`

N

Count

Discussion
Performs the operation

$$D_{nM} = A_{nI} \bullet B + C \qquad \text{n} = \{0, \text{N-1}\}$$

Multiplies vector `A` by scalar `B` and then adds scalar `C` to each product. Results are stored in vector `D`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsmsaD

Vector scalar multiply and scalar add; double precision.

```
void vDSP_vsmsaD (double * A,
vDSP_Stride I,
double * B,
double * C,
double * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Double-precision real input vector

I

Stride for `A`

B

Double-precision real input scalar

C

Double-precision real input scalar

D

Double-precision real output vector

L

Stride for `D`

N

Count

Discussion
Performs the operation

$$D_{nM} = A_{nI} \bullet B + C \qquad \text{n} = \{0, \text{N-1}\}$$

Multiplies vector `A` by scalar `B` and then adds scalar `C` to each product. Results are stored in vector `D`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsmsb

Vector scalar multiply and vector subtract; single precision.

```
void vDSP_vsmsb (float * A,
vDSP_Stride I,
float * B,
float * C,
vDSP_Stride K,
float * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Single-precision real input vector

I

Stride for `A`

B

Single-precision real input scalar

C

Single-precision real input vector

K

Stride for `C`

D

Single-precision real output vector

L

Stride for `D`

N

Count

Discussion
Performs the operation

$$D_{nM} = A_{nI} \bullet B - C_{nK} \qquad n = \{0, N\text{-}1\}$$

Multiplies vector `A` by scalar `B` and then subtracts vector `C` from the products. Results are stored in vector `D`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsmsbD

Vector scalar multiply and vector subtract; double precision.

```
void vDSP_vsmsbD (double * A,
vDSP_Stride I,
double * B,
double * C,
vDSP_Stride K,
double * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Double-precision real input vector

I

Stride for `A`

B

Double-precision real input scalar

C

Double-precision real input vector

K

Stride for `C`

D

Double-precision real output vector

L

Stride for `D`

N

Count

Discussion
Performs the operation

$$D_{nM} = A_{nI} \bullet B - C_{nK} \qquad n = \{0, N\text{-}1\}$$

Multiplies vector `A` by scalar `B` and then subtracts vector `C` from the products. Results are stored in vector `D`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsmul

Multiplies vector `signal1` by scalar `signal2` and leaves the result in vector `result`; single precision.

```
void vDSP_vsmul (const float input1[],
vDSP_Stride stride1,
const float * input2,
float result[],
vDSP_Stride strideResult,
vDSP_Length size);
```

Discussion
This performs the operation

$$C_{nK} = A_{nI} \bullet B \qquad n = \{0, N\text{-}1\}$$

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_vsmulD

Multiplies vector `signal1` by scalar signal2 and leaves the result in vector `result`; double precision.

```
void vDSP_vsmulD (const double input1[],
vDSP_Stride stride1,
const double * input2,
double result[],
vDSP_Stride strideResult,
vDSP_Length size);
```

Discussion
This performs the operation

$$C_{nK} = A_{nI} \bullet B \qquad n = \{0, N\text{-}1\}$$

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_zvdiv

Complex vector divide; single precision.

```
void vDSP_zvdiv (DSPSplitComplex * A,
vDSP_Stride I,
DSPSplitComplex * B,
vDSP_Stride J,
DSPSplitComplex * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Single-precision complex input vector

I

Stride for `A`

B

Single-precision complex input vector

J

Stride for `B`

C

Single-precision complex output vector

K

Stride for `C`

N

Count

Discussion

Divides vector `B` by vector `A`.

$$C_{nK} = \frac{B_{nJ}}{A_{nI}} \qquad \text{n} = \{0, \text{N-1}\}$$

Availability

Available in Mac OS X v10.4 and later.

Declared In

`vDSP.h`

vDSP_zvdivD

Complex vector divide; double precision.

```
void vDSP_zvdivD (DSPDoubleSplitComplex * A,
vDSP_Stride I,
DSPDoubleSplitComplex * B,
vDSP_Stride J,
DSPDoubleSplitComplex * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Double-precision complex input vector

I

Stride for `A`

B

Double-precision complex input vector

J

Stride for `B`

C

Double-precision complex output vector

K

Stride for `C`

N

Count

Discussion

Divides vector `B` by vector `A`.

$$C_{nK} = \frac{B_{nJ}}{A_{nI}} \qquad \text{n} = \{0, \text{N-1}\}$$

Availability

Available in Mac OS X v10.4 and later.

Declared In

`vDSP.h`

vDSP_zvsma

Complex vector scalar multiply and add; single precision.

```
void vDSP_zvsma (DSPSplitComplex * A,
vDSP_Stride I,
DSPSplitComplex * B,
DSPSplitComplex * C,
vDSP_Stride K,
DSPSplitComplex * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Single-precision complex input vector

I

Stride for `A`

B

Single-precision complex input scalar

C

Single-precision real input vector

K

Stride for `C`

D

Single-precision real output vector

L

Stride for `C`

N

Count

Discussion

Multiplies vector `A` by scalar `B` and add the products to vector `C`. The result is stored in vector `D`.

$$D_{nL} = A_{nI}B + C_{nK}$$

Availability

Available in Mac OS X v10.4 and later.

Declared In

`vDSP.h`

vDSP_zvsmaD

Complex vector scalar multiply and add; double precision.

```
void vDSP_zvsmaD (DSPDoubleSplitComplex * A,
vDSP_Stride I,
DSPDoubleSplitComplex * B,
DSPDoubleSplitComplex * C,
vDSP_Stride K,
DSPDoubleSplitComplex * D,
vDSP_Stride L,
vDSP_Length N);
```

Parameters

A

Double-precision complex input vector

I

Stride for `A`

B

Double-precision complex input scalar

C

Double-precision real input vector

K

Stride for `C`

D

Double-precision real output vector

L

Stride for `C`

N

Count

Discussion

Multiplies vector `A` by scalar `B` and add the products to vector `C`. The result is stored in vector `D`.

$$D_{nL} = A_{nI}B + C_{nK}$$

Availability

Available in Mac OS X v10.4 and later.

Declared In

`vDSP.h`

vDSP_zvzsml

Complex vector multiply by complex scalar; single precision.

```
void vDSP_zvzsml (DSPSplitComplex * A,
vDSP_Stride I,
DSPSplitComplex * B,
DSPSplitComplex * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Single-precision complex input vector

I

Stride for `A`

B

Single-precision complex input scalar

C

Single-precision complex output vector

K

Stride for `C`

N

Count

Discussion
This peforms the operation

$$C_{nK} = A_{nI}B$$

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

vDSP_zvzsmlD

Complex vector multiply by complex scalar; double precision.

```
void vDSP_zvzsml (DSPSplitComplex * A,
vDSP_Stride I,
DSPSplitComplex * B,
DSPSplitComplex * C,
vDSP_Stride K,
vDSP_Length N);
```

Parameters

A

Double-precision complex input vector

I

Stride for `A`

B

Double-precision complex input scalar

C

Double-precision complex output vector

K

Stride for `C`

N

Count

Discussion
This peforms the operation

$$C_{nK} = A_{nI}B$$

Availability
Available in Mac OS X v10.4 and later.

Declared In
`vDSP.h`

Document Revision History

This table describes the changes to *vDSP Vector Scalar Arithmetic Operations Reference*.

Date	Notes
2007-06-15	New document that describes the C API for the vDSP functions that perform arithmetic operations combining a scalar with each element of a vector

Document Revision History

Index

V