

Fortran 浮点数据的 simd 扩展手册

一、主要内容

1. 类型的扩展，扩展了两种新的数据类型。
2. 主要内部函数详细说明。

二、类型的扩展

vector128: 4*32, 4 个 float 数据所组成的向量
vector256: 4*64, 4 个 double 数据所组成的向量

三、内部函数的扩展

VRav: 第一个向量参数
VRbv: 第二个向量参数
VRcv: 第三个向量参数
VRc: 向量返回值
Rbv: 无符号长整型变量，一般用于表示地址
Va: 一段内存地址中的数据 f

1、装入/存储函数

宏定义	指令	操作	参数说明	
			VRav	Rbv
simd_cmplx	vinsf	赋值	扩展类型	四个浮点立即数
simd_load	vlds/vldd	装入	扩展类型	数组或者地址
simd_store	vsts/vstd	存储	扩展类型	数组或者地址
simd_loadu	vlds_ul/vlds_uh vldd_ul/vldd_uh	装入	扩展类型	数组或者地址
simd_storeu	vsts_ul/vsts_uh vstd_ul/vstd_uh	存储	扩展类型	数组或者地址

SIMD_CMPLX

对应指令: vinsf

语法: `result = SIMD_CMPLX(f0, f1, f2, f3);`

参数说明:

fi: 浮点立即数

返回值:

vector128/vector256 类型

功能说明: 把四个立即数赋给浮点向量 `result`。

SIMD_LOAD

对应指令: `vlds/vldd`

语法: `CALL SIMD_LOAD(I, A)`

参数说明:

I: vector128/vector256 类型

A: 数组或者地址

返回值:

-

功能说明: 从 A 起的地址装入 128/256 位数据到向量寄存器中。

例如:

(1) VECTOR128 VF

`REAL*4::A(4)/1.1, 2.1, 3.1, 4.1/`

`CALL SIMD_LOAD(VF, A(1))`

(2) VECTOR256 VD

`REAL*8::A(4)/1.1, 2.1, 3.1, 4.1/`

`CALL SIMD_LOAD(VD, A(1))`

SIMD_LOADU

对应指令: `vlds_ul/vlds_uh, vldd_ul/vldd_uh`

语法: `CALL SIMD_LOADU(I, A)`

参数说明:

I: vector128/vector256 类型

A: 数组或者地址

返回值:

-

功能说明: 从 A 起的地址装入 128/256 位数据到向量寄存器中。

例如:

(1) VECTOR128 VF

`REAL*4::A(4)/1.1, 2.1, 3.1, 4.1/`

CALL SIMD_LOADU(VF, A(3))

(2) VECTOR256 VD

REAL*8::A(4)/1.1, 2.1, 3.1, 4.1/

CALL SIMD_LOADU(VD, A(3))

SIMD_STORE

对应指令: vsts/vstd

语法: CALL SIMD_STORE(I, A)

参数说明:

I: vector128/vector256 类型

A: 数组或者地址

返回值:

无

功能说明: 将向量寄存器中的数据存储在从 A 起的内存中。

例如:

(1) VECTOR128 VF

REAL*4::A(4)

...

...

CALL SIMD_STORE(VF, A(1))

(2) VECTOR256 VD

REAL*8::A(4)

...

...

CALL SIMD_STORE(VD, A(1))

SIMD_STOREU

对应指令: vsts_ul/vsts_uh, vstd_ul/

语法: CALL SIMD_STOREU(I, A)

参数说明:

I: vector128/vector256 类型

A: 数组或者地址

返回值:

无

功能说明: 将向量寄存器中的数据存储在从 A 起的内存中。

例如:

(1) VECTOR128 VF
 REAL*4::A(4)
 ...
 ...
 CALL SIMD_STOREU(VF, A(3))

(2) VECTOR256 VD
 REAL*8::A(4)
 ...
 ...
 CALL SIMD_STOREU(VD, A(3))

2、数据整理函数

内部函数	指令	操作	参数说明			
			VRc (返回值)	VRav	VRbv	VRcv
simd_vinsfn n=(0,1,2,3)	vinsf	浮点向量插入	vector128/ vector256	vector128/ vector256	vector128/ vector256	/
simd_vextfn n=(0,1,2,3)	vextf	浮点向量提取	vector128/ vector256	vector128/ vector256	/	/
simd_vcopyf	vcpyf	浮点向量拷贝	vector128 vector256	vector128 vector256		

SIMD_VINSFn

对应指令：vinsf

语法：result = SIMD_VINSFn(x,y)

参数说明：

x: vector128 类型/vector256 类型

y: 与 x 的数据类型相同

返回值：

与 x 的数据类型相同

功能说明：

将浮点寄存器 x 低 64 位浮点数据替代 y 中由 n (有效值为 1~4) 指定的 64 位浮点元素，组成并返回新的浮点向量。

该接口实际上对应 4 个接口，分别是 simd_vinsf0、simd_vinsf1、simd_vinsf2、simd_vinsf3，比如：

例如：

VECTOR256 X, Y, Z

...
 ...
 Z=SIMD_VINSF1(X,Y)

SIMD_VEXTFn

对应指令: vextf

语法: result = SIMD_VEXTFn(x)

参数说明:

x: vector128/vector256 类型

y: 与 x 的数据类型相同

返回值:

与 x 的数据类型相同

功能说明:

将 x 中由 n (有效值为 1~4) 指定的 S 浮点或 D 浮点元素存入目的寄存器的 <63:0> 中。此指令可实现 S 浮点和 D 浮点的提取。

该接口实际上对应 4 个接口, 分别是 simd_vextf0、simd_vextf1、simd_vextf2、simd_vextf3。

SIMD_VCPYF

对应指令: vcpyf

语法: result = SIMD_VCPYF(x)

参数说明:

x: vector128/vector256 类型

返回值:

与 x 的数据类型相同

功能说明:

将 x 低 64 位浮点数据, 复制成 4 个相同元素, 组成新的浮点向量写入目的寄存器。

5. 浮点运算函数

内部函数	指令	操作	参数说明			
			VRc (返回值)	VRav	VRbv	VRcv
simd_vadds	vadds	+	vector128	vector128	vector128	/
simd_vsubs	vsubs	-	vector128	vector128	vector128	/
simd_vmuls	vmuls	*	vector128	vector128	vector128	/
simd_vmas	vmas	乘加	vector128	vector128	vector128	vector128
simd_vmss	vmss	乘减	vector128	vector128	vector128	vector128

simd_vnmas	vnmas	负乘加	vector128	vector128	vector128	vector128
simd_vnmss	vnmss	负乘减	vector128	vector128	vector128	vector128
simd_vaddd	vaddd	+	vector256	vector256	vector256	/
simd_vsubd	vsubd	-	vector256	vector256	vector256	/
simd_vmuld	vmuld	*	vector256	vector256	vector256	/
simd_vmad	vmad	乘加	vector256	vector256	vector256	vector256
simd_vmsd	vmsd	乘减	vector256	vector256	vector256	vector256
simd_vnmad	vnmad	负乘加	vector256	vector256	vector256	vector256
simd_vnmsd	vnmsd	负乘减	vector256	vector256	vector256	vector256
simd_vseleq	vseleq	条件选择	扩展浮点	扩展浮点	扩展浮点	扩展浮点
simd_vselne	vselne		扩展浮点	扩展浮点	扩展浮点	扩展浮点
simd_vsellt	vsellt		扩展浮点	扩展浮点	扩展浮点	扩展浮点
simd_vselle	vselle		扩展浮点	扩展浮点	扩展浮点	扩展浮点
simd_vselgt	vselgt		扩展浮点	扩展浮点	扩展浮点	扩展浮点
simd_vselge	vselge		扩展浮点	扩展浮点	扩展浮点	扩展浮点
simd_vcpys	vc pys	符号拷贝	扩展浮点	扩展浮点	扩展浮点	/
simd_vcpyse	vc pyse		扩展浮点	扩展浮点	扩展浮点	/
simd_vcpysn	vc pynsn		扩展浮点	扩展浮点	扩展浮点	/
simd_vdivs	vdivs	除法	vector128	vector128	vector128	—
simd_vdivd	vdivd	除法	vector256	vector256	vector256	—
simd_vsqrts	vsqrts	求平方根	vector128	vector128	—	—
simd_vsqrtd	vsqrtd	求平方根	vector256	vector256	—	—
simd_vfcmp eq	vc mp eq	等于比较	vector128/ vector256	vector128/ vector256	vector128/ vector256	—
simd_vfcmpl e	vc mpl e	小于等于比较	vector128/ vector256	vector128/ vector256	vector128/ vector256	—
simd_vfcmpl t	vc mpl t	小于比较	vector128/ vector256	vector128/ vector256	vector128/ vector256	—
simd_vfcmp un	vc mp un	无序比较	vector128/ vector256	vector128/ vector256	vector128/ vector256	—

SIMD_VADDS

对应指令：vadds

语法：result = SIMD_VADDS(x,y)

参数说明：

x: vector128 类型

y: vector128 类型

返回值:

vector128 类型

功能说明:

将 x 和 y 中的 4 个 float 类型浮点分别相加。

SIMD_VADDD

对应指令: vaddd

语法: result = SIMD_VADDD(x,y)

参数说明:

x: vector256 类型

y: vector256 类型

返回值:

vector256 类型

功能说明:

将 x 和 y 中的 4 个 double 类型浮点分别相加。

SIMD_VSUBS

对应指令: vsubs

语法: result = SIMD_VSUBS(x,y)

参数说明:

x: vector128 类型

y: vector128 类型

返回值:

vector128 类型

功能说明:

将 x 和 y 中的 4 个 float 类型浮点分别相减。

SIMD_VSUBD

对应指令: vsubd

语法: result = SIMD_VSUBD(x,y)

参数说明:

x: vector256 类型

y: vector256 类型

返回值:

vector256 类型

功能说明:

将 x 和 y 中的 4 个 double 类型浮点分别相减。

SIMD_VMULS

对应指令: vmuls

语法: result = SIMD_VMULS(x,y)

参数说明:

x: vector128 类型

y: vector128 类型

返回值:

vector128 类型

功能说明:

将 x 和 y 中的 4 个 float 类型浮点分别相乘。

SIMD_VMULD

对应指令: vmuld

语法: result = SIMD_VMULD(x,y)

参数说明:

x: vector256 类型

y: vector256 类型

返回值:

vector256 类型

功能说明:

将 x 和 y 中的 4 个 double 类型浮点分别相乘。

SIMD_VDIVS

对应指令: vdivs

语法: result = SIMD_VDIVS(x,y)

参数说明:

x: vector128 类型

y: vector128 类型

返回值:

vector128 类型

功能说明:

将 x 和 y 中的 4 个 float 类型浮点分别相除。

SIMD_VDIVD

对应指令: vdivd

语法: result = SIMD_VDIVD(x,y)

参数说明:

x: vector256 类型

y: vector256 类型

返回值:

vector256 类型

功能说明:

将 x 和 y 中的 4 个 double 类型浮点分别相除。

SIMD_VSQRTS

对应指令: vsqrts

语法: result = SIMD_VSQRTD(x)

参数说明:

x: vector128 类型

返回值:

vector128 类型

功能说明:

将 x 中的 4 个 float 类型浮点分别进行求平方根运算。

SIMD_VSQRTD

对应指令: vsqrtf

语法: result = SIMD_VSQRTD(x)

参数说明:

x: vector256 类型

返回值:

vector256 类型

功能说明:

将 x 中的 4 个 double 类型浮点分别进行求平方根运算。

SIMD_VMAS

对应指令: vmas

语法: result = SIMD_VMAS(x,y,z)

参数说明:

x: vector128 类型

y: vector128 类型

z: vector128 类型

返回值:

vector128 类型

功能说明:

vector128 类型的乘加运算, 将 x 中的 4 个 float 浮点数和 y 中的 4 个 float 浮点数以及 z 中的两个浮点数分别进行乘加运算。

SIMD_VMAD

对应指令: vmad

语法: result = SIMD_VMAD(x,y,z)

参数说明:

x: vector256 类型

y: vector256 类型

z: vector256 类型

返回值:

vector256 类型

功能说明:

vector256 类型的乘加运算, 将 x 中的 4 个 double 浮点数和 y 中的 4 个 double 浮点数以及 z 中的 4 个浮点数分别进行乘加运算。

SIMD_VMSS

对应指令: vmss

语法: result = SIMD_VMSS(x,y,z)

参数说明:

x: vector128 类型

y: vector128 类型

z: vector128 类型

返回值:

vector128 类型

功能说明:

vector128 类型的乘减运算, 将 x 中的 4 个 float 浮点数和 y 中的 4 个 float 浮点数以及 z 中的两个浮点数分别进行乘减运算。

SIMD_VMSD

对应指令: vmsd

语法: result = SIMD_VMSD(x,y,z)

参数说明:

x: vector256 类型

y: vector256 类型

z: vector256 类型

返回值:

vector256 类型

功能说明:

vector256 类型的乘减运算, 将 x 中的 4 个 double 浮点数和 y 中的 4 个 double 浮点数以及 z 中的 4 个浮点数分别进行乘减运算。

SIMD_VNMAS

对应指令: vnmas

语法: result = SIMD_VNMAS(x,y,z)

参数说明:

x: vector128 类型

y: vector128 类型

z: vector128 类型

返回值:

vector128 类型

功能说明:

vector128 类型的负乘加运算, 将 x 中的 4 个 float 浮点数和 y 中的 4 个 float 浮点数以及 z 中的两个浮点数分别进行负乘加运算。

SIMD_VNMAD

对应指令: vnmad

语法: result = SIMD_VNMAD(x,y,z)

参数说明:

x: vector256 类型

y: vector256 类型

z: vector256 类型

返回值:

vector256 类型

功能说明:

vector256 类型的负乘加运算, 将 x 中的 4 个 double 浮点数和 y 中的 4 个 double 浮点数以及 z 中的两个浮点数分别进行负乘加运算。

SIMD_VNMSS

对应指令: vnmss

语法: result = SIMD_VNMSS(x,y,z)

参数说明:

x: vector128 类型

y: vector128 类型

z: vector128 类型

返回值:

vector128 类型

功能说明:

vector128 类型的负乘减运算, 将 x 中的 4 个 float 浮点数和 y 中的 4 个 float 浮点数以及 z 中的两个浮点数分别进行负乘减运算。

SIMD_VNMSD

对应指令: vnmsd

语法: result = SIMD_VNMSD(x,y,z)

参数说明:

x: vector256 类型

y: vector256 类型

z: vector256 类型

返回值:

vector256 类型

功能说明:

vector256 类型的负乘减运算, 将 x 中的 4 个 double 浮点数和 y 中的 4 个 double 浮点数以及 z 中的两个浮点数分别进行负乘减运算。

SIMD_VFCMPEQ

对应指令: vfcmpeq

语法: result = SIMD_VFCMPEQ(x,y)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

将x和y中的对应浮点向量元素进行等于比较, 如果条件成立, 则将非“0”浮点值(2.0)写入result的对应位置, 否则将真“0”写入result的对应位置。

SIMD_VFCMPLE

对应指令: vfcmple

语法: result = SIMD_VFCMPLE (x,y)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

将x和y中的对应浮点向量元素进行小于等于比较, 如果条件成立, 则将非“0”浮点值(2.0)写入result的对应位置, 否则将真“0”写入result的对应位置。

SIMD_VFCMPLT

对应指令: vfcmplt

语法: `result = SIMD_VFCMPLT(x,y)`

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

将x和y中的对应浮点向量元素进行小于比较, 如果条件成立, 则将非“0”浮点值(2.0)写入result的对应位置, 否则将真“0”写入result的对应位置。

SIMD_VFCMPUN

对应指令: `vfcmpun`

语法: `result = SIMD_VFCMPUN(x,y)`

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

将x和y中的对应浮点向量元素进行无序比较, 如果条件成立, 则将非“0”浮点值(2.0)写入result的对应位置, 否则将真“0”写入result的对应位置。

SIMD_VSELEQ

对应指令: `vseleq`

语法: `result = SIMD_VSELEQ(x,y,z)`

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

z: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

判断 vector256/vector128 类型的参数 x, 如果等于 0, 则返回 y 的值; 否则返回 z 的值。

SIMD_VSELNE

对应指令: `vselne`

语法: `result = SIMD_VSELNE(x,y,z)`

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

z: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

判断 vector256/vector128 类型的参数 x, 如果不等于 0, 则返回 y 的值; 否则返回 z 的值。

SIMD_VSELLT

对应指令: vsellt

语法: result = SIMD_VSELLT(x,y,z)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

z: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

判断 vector256/vector128 类型的参数 x, 如果小于 0, 则返回 y 的值; 否则返回 z 的值。

SIMD_VSELLE

对应指令: vsellt

语法: result = SIMD_VSELLE(x,y,z)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

z: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

判断 vector256/vector128 类型的参数 x, 如果小于等于 0, 则返回 y 的值; 否则返回 z 的值。

SIMD_VSELGT

对应指令: vselgt

语法: result = SIMD_VSELGT(x,y,z)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

z: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

判断 vector256/vector128 类型的参数 x, 如果大于 0, 则返回 y 的值; 否则返回 z 的值。

SIMD_VSELGE

对应指令: vselge

语法: result = SIMD_VSELGE(x,y,z)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

z: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

判断 vector256/vector128 类型的参数 x, 如果大于等于 0, 则返回 y 的值; 否则返回 z 的值。

SIMD_VCPYS

对应指令: vcpys

语法: result = SIMD_VCPYS(x,y)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

浮点拷贝符号, 返回值中的符号位是 x 的符号位, 其余部分为 y 中的数据。

SIMD_VCPYSE

对应指令: vcpyse

语法: result = SIMD_VCPYSE(x,y)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

浮点拷贝符号和指数, 返回值中的符号位和指数位是 x 的数据, 其余部分为 y 中的数据。

SIMD_VCPYSN

对应指令: vcpysn

语法: result = SIMD_VCPYSN(x,y)

参数说明:

x: vector128/vector256 类型

y: vector128/vector256 类型

返回值:

vector128/vector256 类型

功能说明:

浮点拷贝符号反码, 返回值中的符号位是 x 的符号位的反码, 其余部分为 y 中的数据。