

Larrabee New Instruction (LRBni)

- [Load/Store](#) (page 1)
- [Gather/Scatter](#) (page 2)
- [Logical/Bits](#) (page 2)
- [Rotation/Shift/Shuffle](#) (page 6)
- [Convert/Fixup](#) (page 7)
- [Arithmetic](#) (page 8)
- [Compress/Expand](#) (page 13)
- [Utility](#) (page 14)
- [Math](#) (page 15)
- [Scalar](#) (page 19)
- [Prefetch](#) (page 20)
- [Vector Mask](#) (page 20)

	Return type	Arguments	description
Load/Store			
_mm512_upconv_int32	_M512I	void *v, _MM_UPCONV_I32_ENUM s, _MM_MEM_HINT_ENUM	Load from memory and upconvert to int32 vector
_mm512_upconv_float32	_M512	void *v, _MM_UPCONV_F32_ENUM s, _MM_MEM_HINT_ENUM	Load from memory and upconvert to float32 vector
_mm512_upconv_int64	_M512I	void *v, _MM_UPCONV_I64_ENUM s, _MM_MEM_HINT_ENUM	Load from memory and upconvert to int64 vector
_mm512_upconv_float64	_M512D	void *v, _MM_UPCONV_F64_ENUM s, _MM_MEM_HINT_ENUM	Load from memory and upconvert to float64 vector
_mm512_loadd	_M512	void *m, _MM_FULLLUP32_ENUM full_up, _MM_BROADCAST32_ENUM broadcast, _MM_MEM_HINT_ENUM	Load doubleword vector
_mm512_mask_loadd	_M512	_M512 v1_old, _mmask k1, void *m, _MM_FULLLUP32_ENUM full_up, _MM_BROADCAST32_ENUM broadcast, _MM_MEM_HINT_ENUM	Load doubleword vector under mask
_mm512_loadq	_M512	void *m, _MM_FULLLUP64_ENUM, _MM_BROADCAST64_ENUM broadcast, _MM_MEM_HINT_ENUM	Load quadword vector
_mm512_mask_loadq	_M512	_M512 v1_old, _mmask k1, void *m, _MM_FULLLUP64_ENUM, _MM_BROADCAST64_ENUM broadcast, _MM_MEM_HINT_ENUM	Load quadword vector under mask
_mm512_stored	void	void *m, _M512 v2, _MM_DOWNCONV32_ENUM down_conv, _MM_STORE_SUBSET32_ENUM subset, _MM_MEM_HINT_ENUM	Store doubleword vector
_mm512_mask_stored	void	void *m, _mmask k1, _M512 v2, _MM_DOWNCONV32_ENUM down_conv, _MM_STORE_SUBSET32_ENUM subset, _MM_MEM_HINT_ENUM	Store doubleword vector under mask
_mm512_storeq	void	void *m, _M512 v2, _MM_DOWNCONV64_ENUM, _MM_STORE_SUBSET64_ENUM subset, _MM_MEM_HINT_ENUM	Store quadword vector
_mm512_mask_storeq	Void	void *m, _mmask k1, _M512 v2, _MM_DOWNCONV64_ENUM, _MM_STORE_SUBSET64_ENUM subset,	Store quadword vector under mask

		MM MEM HINT ENUM	
Gather/Scatter			
<code>_mm512_mask_gatherd</code>	<code>_M512</code>	<code>_M512I v1_old,</code> <code>__mmask k1,</code> <code>_M512I index,</code> <code>void *m,</code> <code>_MM_FULLLUP32_ENUM up_conv,</code> <code>_MM_INDEX_SCALE_ENUM scale,</code> <code>MM MEM HINT ENUM</code>	Gather all elements vector under mask
<code>_mm512_gatherd</code>	<code>_M512</code>	<code>_M512I index,</code> <code>void *m,</code> <code>_MM_FULLLUP32_ENUM up_conv,</code> <code>_MM_INDEX_SCALE_ENUM scale,</code> <code>MM MEM HINT ENUM</code>	Gather all elements vector
<code>_mm512_gatherpfd</code>	<code>void</code>	<code>_M512I, void *,</code> <code>_MM_FULLLUP32_ENUM,</code> <code>_MM_INDEX_SCALE_ENUM,</code> <code>MM MEM HINT ENUM</code>	
<code>_mm512_mask_gatherpfd</code>	<code>void</code>	<code>_M512I,</code> <code>__mmask, void *,</code> <code>_MM_FULLLUP32_ENUM,</code> <code>_MM_INDEX_SCALE_ENUM,</code> <code>MM MEM HINT ENUM</code>	
<code>_mm512_scatterd</code>	<code>void</code>	<code>void *m,</code> <code>_M512I index,</code> <code>_M512 v1,</code> <code>_MM_DOWNCONV32_ENUM down_conv,</code> <code>_MM_INDEX_SCALE_ENUM scale,</code> <code>MM MEM HINT ENUM</code>	Scatter element vector
<code>_mm512_mask_scatterd</code>	<code>void</code>	<code>void *m, __mmask k1,</code> <code>_M512I index,</code> <code>_M512 v1,</code> <code>_MM_DOWNCONV32_ENUM down_conv,</code> <code>_MM_INDEX_SCALE_ENUM scale,</code> <code>MM MEM HINT ENUM</code>	Scatter element vector under mask
<code>_mm512_scatterpfd</code>	<code>void</code>	<code>void *, _M512I,</code> <code>_MM_DOWNCONV32_ENUM,</code> <code>_MM_INDEX_SCALE_ENUM,</code> <code>MM MEM HINT ENUM</code>	Scatter prefetch element vector
<code>_mm512_mask_scatterpfd</code>	<code>void</code>	<code>void *,</code> <code>__mmask,</code> <code>_M512I,</code> <code>_MM_DOWNCONV32_ENUM,</code> <code>_MM_INDEX_SCALE_ENUM,</code> <code>MM MEM HINT ENUM</code>	Scatter prefetch element vector under mask
Logical/Bits			
<code>_mm512_mask_test_pi</code>	<code>__mmask</code>	<code>__mmask k1,</code> <code>_M512I v1,</code> <code>_M512I v2</code>	Logical AND and set vector mask under mask
<code>_mm512_test_pi</code>	<code>__mmask</code>	<code>_M512I v1,</code> <code>_M512I v2</code>	Logical AND and set vector mask
<code>_mm512_xor_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise XOR int32 vectors
<code>_mm512_mask_xor_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>__mmask k1,</code> <code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise XOR int32 vectors under mask
<code>_mm512_xor_pq</code>	<code>M512I</code>	<code>M512I v2, M512I v3</code>	Bitwise XOR int64 vectors
<code>_mm512_mask_xor_pq</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>__mmask k1,</code> <code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise XOR int64 vectors under mask
<code>_mm512_or_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise OR int32 vectors
<code>_mm512_mask_or_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>__mmask k1,</code> <code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise OR int32 vectors under mask
<code>_mm512_or_pq</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise OR int64 vectors
<code>_mm512_mask_or_pq</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>__mmask k1,</code> <code>_M512I v2,</code> <code>_M512I v3</code>	Bitwise OR int64 vectors under mask

<code>mm512_andn_pi</code>	<code>M512I</code>	<code>M512I v2, M512I v3</code>	Bitwise AND NOT int32 vectors
<code>_mm512_mask_andn_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Bitwise AND NOT int32 vectors under mask
<code>_mm512_andn_pq</code>	<code>_M512I</code>	<code>M512I v2,</code> <code>M512I v3</code>	Bitwise AND NOT int64 vectors
<code>_mm512_mask_andn_pq</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Bitwise AND NOT int64 vectors under mask
<code>_mm512_and_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	Bitwise AND int32 vectors
<code>_mm512_mask_and_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Bitwise AND int32 vectors under mask
<code>_mm512_and_pq</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	Bitwise AND int64 vectors
<code>_mm512_mask_and_pq</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Bitwise AND int64 vectors under mask
<code>_mm512_bitinterleave11_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	1:1 bit-interleave int32 vectors
<code>_mm512_mask_bitinterleave11_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	1:1 bit-interleave int32 vectors under mask
<code>_mm512_bitinterleave21_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	2:1 bit-interleave int32 vectors
<code>_mm512_mask_bitinterleave21_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	2:1 bit-interleave int32 vectors under mask
<code>_mm512_clampz_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	Clamp to zero int32 vector
<code>_mm512_mask_clampz_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Clamp to zero int32 vector under mask
<code>_mm512_clampz_ps</code>	<code>_M512</code>	<code>_M512 v2,</code> <code>M512 v3</code>	Clamp to zero float32 vector
<code>_mm512_mask_clampz_ps</code>	<code>_M512</code>	<code>_M512 v1_old,</code> <code>_mmask k1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Clamp to zero float32 vector under mask
<code>_mm512_cmpbmskeq_pu</code>	<code>__mmask</code>	<code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask
<code>_mm512_mask_cmpbmskeq_pu</code>	<code>__mmask</code>	<code>_mmask k1,</code> <code>M512I v1,</code> <code>M512I v2,</code> <code>int field</code>	Compare bytemasked uint32 vectors and set mask under mask
<code>_mm512_cmpbmsklt_pu</code>	<code>__mmask</code>	<code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask
<code>_mm512_mask_cmpbmsklt_pu</code>	<code>__mmask</code>	<code>_mmask k1,</code> <code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask under mask
<code>_mm512_mask_cmpbmskle_pu</code>	<code>__mmask</code>	<code>_mmask k1,</code> <code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask under mask
<code>_mm512_cmpbmskle_pu</code>	<code>__mmask</code>	<code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask
<code>_mm512_mask_cmpbmskneq_pu</code>	<code>__mmask</code>	<code>_mmask k1,</code> <code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask under mask
<code>_mm512_cmpbmskneq_pu</code>	<code>__mmask</code>	<code>M512I v1,</code> <code>M512I v2,</code> <code>MM BMSK FIELD ENUM field</code>	Compare bytemasked uint32 vectors and set mask
<code>mm512_mask_cmpbmsknltpu</code>	<code>mmask</code>	<code>mmask k1,</code>	Compare bytemasked uint32

		_M512I v1, _M512I v2, MM BMSK FIELD ENUM field	vectors and set mask under mask
_mm512_cmpbmsknltpu	__mmask	_M512I v1, _M512I v2, MM BMSK FIELD ENUM field	Compare bytemasked uint32 vectors and set mask
_mm512_mask_cmpbmsknlpu	__mmask	__mmask k1, _M512I v1, _M512I v2, MM BMSK FIELD ENUM field	Compare bytemasked uint32 vectors and set mask under mask
_mm512_cmpbmsknlpu	__mmask	_M512I v1, _M512I v2, MM BMSK FIELD ENUM field	Compare bytemasked uint32 vectors and set mask
_mm512_mask_cmpeqpd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpeqpd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpltpd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpltpd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmplepd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmplepd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpunordpd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpunordpd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpneqpd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpneqpd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpnltpd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpnltpd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpnlpu	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpnlpu	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpordpd	__mmask	__mmask k1, _M512D v1, _M512D v2	Compare float64 vectors and set mask under mask
_mm512_cmpordpd	__mmask	_M512D v1, _M512D v2	Compare float64 vectors and set mask
_mm512_mask_cmpeqpi	__mmask	__mmask k1, _M512I v1, _M512I v2	Compare int32 vectors and set mask under mask
_mm512_cmpeqpi	__mmask	_M512I v1, _M512I v2	Compare int32 vectors and set mask
_mm512_mask_cmpltpi	__mmask	__mmask k1, _M512I v1, _M512I v2	Compare int32 vectors and set mask under mask
_mm512_cmpltpi	__mmask	_M512I v1, _M512I v2	Compare int32 vectors and set mask
_mm512_mask_cmplepi	__mmask	__mmask k1, _M512I v1, _M512I v2	Compare int32 vectors and set mask under mask
_mm512_cmplepi	__mmask	_M512I v1, _M512I v2	Compare int32 vectors and set mask
_mm512_mask_cmpneqpi	__mmask	__mmask k1, _M512I v1, _M512I v2	Compare int32 vectors and set mask under mask
_mm512_cmpneqpi	__mmask	_M512I v1, _M512I v2	Compare int32 vectors and set mask
_mm512_mask_cmpnltpi	__mmask	__mmask k1, _M512I v1,	Compare int32 vectors and set mask under mask

		M512I v2	
_mm512_cmpnle_pu	__mmask	_M512I v1, M512I v2	Compare uint32 vectors and set mask
Rotation/ Shift/ Shuffle			
_mm512_rotatefield_pi	_M512I	_M512I v2, _MM_BITPOSITION32_ENUM rotation, _MM_BITPOSITION32_ENUM bit_idx_low, _MM_BITPOSITION32_ENUM bit_idx_high	Rotate and bitfield-mask int32 vector
_mm512_mask_rotatefield_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, _MM_BITPOSITION32_ENUM rotation, _MM_BITPOSITION32_ENUM bit_idx_low, _MM_BITPOSITION32_ENUM bit_idx_high	Rotate and bitfield-mask int32 vector under mask
_mm512_insertfield_pi	_M512I	_M512I v2, _M512I v3, _MM_BITPOSITION32_ENUM rotation, _MM_BITPOSITION32_ENUM bit_idx_low, _MM_BITPOSITION32_ENUM bit_idx_high	Rotate int32 vector and bitfield-insert into int32 vector
_mm512_mask_insertfield_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, _M512I v3, _MM_BITPOSITION32_ENUM rotation, _MM_BITPOSITION32_ENUM bit_idx_low, _MM_BITPOSITION32_ENUM bit_idx_high	Rotate int32 vector and bitfield-insert into int32 vector under mask
_mm512_sll_pi	M512I	M512I v2, M512I v3	Shift int32 vector left logical
_mm512_mask_sll_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, M512I v3	Shift int32 vector left logical under mask
_mm512_sra_pi	_M512I	_M512I v2, M512I v3	Shift int32 vector right arithmetic
_mm512_mask_sra_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, M512I v3	Shift int32 vector right arithmetic under mask
_mm512_srl_pi	_M512I	_M512I v2, M512I v3	Shift int32 vector right logical
_mm512_mask_srl_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, M512I v3	Shift int32 vector right logical under mask
_mm512_shuf128x32	_M512	M512 v2, _MM_PERM_ENUM perm128, _MM_PERM_ENUM perm32	Shuffle vector dqwords then doublewords
_mm512_mask_shuf128x32	_M512	_M512 v1_old, __mmask k1, _M512 v2, _MM_PERM_ENUM perm128, _MM_PERM_ENUM perm32	Shuffle vector dqwords then doublewords under mask
_mm512_shuf128x32_m	_M512	void *m, _MM_PERM_ENUM perm128, _MM_PERM_ENUM perm32, _MM_MEM_HINT_ENUM	Shuffle vector dqwords then doublewords from memory
_mm512_mask_shuf128x32_m	_M512	_M512 v1_old, __mmask k1, void *m, _MM_PERM_ENUM perm128, _MM_PERM_ENUM perm32, _MM_MEM_HINT_ENUM	Shuffle vector dqwords then doublewords from memory under mask
mm512_swizzle_r32	M512	M512 v, _MM_SWIZZLE_ENUM s	Swizzle doubleword vector
mm512_swizzle_r64	M512	M512 v, _MM_SWIZZLE_ENUM s	Swizzle quadword vector
Convert/Fixup			
_mm512_cvtins_ps2f11	_M512	_M512 v1, M512 v2,	Convert and insert float32 vector to float11:11:10 vector

		_MM_ROUND_MODE_ENUM rc, _MM_FLOAT11_FIELD_ENUM field	
_mm512_mask_cvtins_ps2f11	_M512	_M512 v1, _mmask k1, _M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_FLOAT11_FIELD_ENUM field	Convert and insert float32 vector to float11:11:10 vector under mask
_mm512_cvtins_ps2u10	_M512	_M512 v1, _M512 v2, _MM_UNORM10_FIELD_ENUM field	Convert and insert float32 vector to unorm10:10:10:2 vector
_mm512_mask_cvtins_ps2u10	_M512	_M512 v1, _mmask k1, _M512 v2, _MM_UNORM10_FIELD_ENUM field	Convert and insert float32 vector to unorm10:10:10:2 vector under mask
_mm512_cvtl_pd2pi	_M512I	_M512I v1_old, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to lower half of int32 vector
_mm512_mask_cvtl_pd2pi	_M512I	_M512I v1_old, _mmask k1, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to lower half of int32 vector under mask
_mm512_cvth_pd2pi	_M512I	_M512I v1_old, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to higher half of int32 vector
_mm512_mask_cvth_pd2pi	_M512I	_M512I v1_old, _mmask k1, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to higher half of int32 vector under mask
_mm512_cvtl_pd2ps	_M512	_M512 v1_old, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to lower half of float32 vector
_mm512_mask_cvtl_pd2ps	_M512	_M512 v1_old, _mmask k1, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to lower half of float32 vector under mask
_mm512_cvth_pd2ps	_M512	_M512 v1_old, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to higher half of float32 vector
_mm512_mask_cvth_pd2ps	_M512	_M512 v1_old, _mmask k1, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to higher half of float32 vector under mask
_mm512_cvtl_pd2pu	_M512I	_M512I v1_old, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to lower half of uint32 vector
_mm512_mask_cvtl_pd2pu	_M512I	_M512I v1_old, _mmask k1, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to lower half of uint32 vector under mask
_mm512_cvth_pd2pu	_M512I	_M512I v1_old, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to higher half of uint32 vector
_mm512_mask_cvth_pd2pu	_M512I	_M512I v1_old, _mmask k1, _M512D v2, _MM_ROUND_MODE_ENUM rc	Convert float64 vector to higher half of uint32 vector under mask
_mm512_cvtl_pi2pd	_M512D	_M512I v2	Convert lower half of int32 vector to float64 vector
_mm512_mask_cvtl_pi2pd	_M512D	_M512D v1_old, _mmask k1, _M512I v2	Convert lower half of int32 vector to float64 vector under mask
_mm512_cvth_pi2pd	_M512D	_M512I v2	Convert higher half of int32 vector to float64 vector
_mm512_mask_cvt_pi2pd	_M512D	_M512D v1_old, _mmask k1, _M512I v2	Convert higher half of int32 vector to float64 vector under mask
_mm512_cvt_pi2ps	_M512	_M512I v2, _MM_EXP_ADJ_ENUM expadj	Convert int32 vector to float32 vector
_mm512_mask_cvt_pi2ps	_M512	_M512 v1_old, _mmask k1, _M512I v2, _MM_EXP_ADJ_ENUM expadj	Convert int32 vector to float32 vector under mask
_mm512_cvtl_ps2pd	_M512D	_M512 v2	Convert lower half of float32 vector to float64 vector
_mm512_mask_cvtl_ps2pd	_M512D	_M512D v1_old, _mmask k1,	Convert lower half of float32 vector to float64 vector under

		M512 v2	mask
_mm512_cvth_ps2pd	_M512D	_M512 v2	Convert higher half of float32 vector to float64 vector
_mm512_mask_cvth_ps2pd	_M512D	_M512D v1_old, _mmask k1, M512 v2	Convert higher half of float32 vector to float64 vector under mask
_mm512_cvt_ps2pi	_M512I	M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_EXP_ADJ_ENUM expadj	Convert float32 vector to int32 vector
_mm512_mask_cvt_ps2pi	_M512I	_M512I v1_old, _mmask k1, M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_EXP_ADJ_ENUM expadj	Convert float32 vector to int32 vector under mask
_mm512_cvt_ps2pu	_M512I	M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_EXP_ADJ_ENUM expadj	Convert float32 vector to uint32 vector
_mm512_mask_cvt_ps2pu	_M512I	_M512I v1_old, _mmask k1, M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_EXP_ADJ_ENUM expadj	Convert float32 vector to uint32 vector under mask
_mm512_cvt_ps2srgb8	_M512	M512 v2	Convert float32 vector to SRGB8 vector
_mm512_mask_cvt_ps2srgb8	_M512	M512 v1_old, _mmask k1, M512 v2	Convert float32 vector to SRGB8 vector under mask
_mm512_cvttl_pu2pd	_M512D	_M512I v2	Convert lower half of uint32 vector to float64 vector
_mm512_mask_cvttl_pu2pd	_M512D	_M512D v1_old, _mmask k1, M512I v2	Convert lower half of uint32 vector to float64 vector under mask
_mm512_cvth_pu2pd	_M512D	_M512I v2	Convert higher half of uint32 vector to float64 vector
_mm512_mask_cvth_pu2pd	_M512D	M512D v1_old, _mmask k1, M512I v2	Convert higher half of uint32 vector to float64 vector under mask
_mm512_cvt_pu2ps	_M512	M512I v2, _MM_EXP_ADJ_ENUM expadj	Convert uint32 vector to float32 vector
_mm512_mask_cvt_pu2ps	_M512	M512 v1_old, _mmask k1, M512I v2, _MM_EXP_ADJ_ENUM expadj	Convert uint32 vector to float32 vector under mask
_mm512_mask_fixup_ps	_M512	M512 v1, _mmask k1, M512 v2, _MM_FIXUPTABLE_ENUM table	Fix up special float32 vector numbers under mask
_mm512_fixup_ps	_M512	M512 v1, M512 v2, _MM_FIXUPTABLE_ENUM table	Fix up special float32 vector numbers
_mm512_getexp_ps	_M512	M512 v2	Extract float32 vector of exponents
_mm512_mask_getexp_ps	_M512	M512 v1_old, _mmask k1, M512 v2	Extract float32 vector of exponents under mask
Arithmetic			
_mm512_adc_pi	_M512I	_M512I v1, _mmask k2, M512I v3, _mmask *carry	Add int32 vectors with carry
_mm512_mask_adc_pi	_M512I	_M512I v1, _mmask k1, _mmask k2, M512I v3, _mmask *carry	Add int32 vectors with carry under mask
_mm512_addn_pd	_M512D	M512D v2, M512D v3	Add and negate float64 vectors
_mm512_mask_addn_pd	_M512D	M512D v1_old, _mmask k1, M512D v2, M512D v3	Add and negate float64 vectors under mask
_mm512_addn_ps	_M512	M512 v2, M512 v3	Add and negate float32 vectors
_mm512_mask_addn_ps	_M512	M512 v1_old, _mmask k1, M512 v2, M512 v3	Add and negate float32 vectors under mask

_mm512_add_pd	_M512D	_M512D v2, M512D v3	Add float64 vectors
_mm512_mask_add_pd	_M512D	_M512D v1_old, _mmask k1, _M512D v2, M512D v3	Add float64 vectors under mask
_mm512_add_pi	_M512I	_M512I v2, M512I v3	Add int32 vectors
_mm512_mask_add_pi	_M512I	_M512I v1_old, _mmask k1, _M512I v2, M512I v3	Add int32 vectors under mask
_mm512_add_ps	_M512	_M512 v2, M512 v3	Add float32 vectors
_mm512_mask_add_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, M512 v3	Add float32 vectors under mask
_mm512_addsetc_pi	_M512I	_M512I v1, _M512I v3, _mmask *carry	Add int32 vectors and set carry
_mm512_mask_addsetc_pi	_M512I	_M512I v1, _mmask k1, _mmask k2_old, _M512I v3, _mmask *carry	Add int32 vectors and set carry under mask
_mm512_addsets_pi	_M512I	_M512I v2, _M512I v3, _mmask *sign	Add int32 vectors and set mask to sign
_mm512_mask_addsets_pi	_M512I	_M512I v1_old, _mmask k1, _M512I v2, _M512I v3, _mmask *sign	Add int32 vectors and set mask to sign under mask
_mm512_addsets_ps	_M512	_M512 v2, _M512 v3, _mmask *sign	Add float32 vectors and set mask to sign
_mm512_mask_addsets_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, _M512 v3, _mmask *sign	Add float32 vectors and set mask to sign under mask
_mm512_madd231_pd	_M512D	_M512D v1, _M512D v2, M512D v3	Multiply and add float64 vectors
_mm512_mask_madd231_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, M512D v3	Multiply and add float64 vectors under mask
_mm512_madd231_pi	_M512I	_M512I v1, _M512I v2, M512I v3	Multiply and add int32 vectors
_mm512_mask_madd231_pi	_M512I	_M512I v1, _mmask k1, _M512I v2, M512I v3	Multiply and add int32 vectors under mask
_mm512_madd231_ps	_M512	_M512 v1, _M512 v2, M512 v3	Multiply and add float32 vectors
_mm512_mask_madd231_ps	_M512	_M512 v1, _mmask k1, _M512 v2, M512 v3	Multiply and add float32 vectors under mask
_mm512_madd233_pi	_M512I	_M512I v2, M512I v3	Multiply and add int32 vectors
_mm512_mask_madd233_pi	_M512I	_M512I v1_old, _mmask k1, _M512I v2, M512I v3	Multiply and add int32 vectors under mask
_mm512_madd233_ps	_M512	_M512 v2, M512 v3	Multiply and add float32 vectors
_mm512_mask_madd233_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, M512 v3	Multiply and add float32 vectors under mask
mm512_maddn132_pd	M512D	M512D v1,	Multiply, add and negate

		_M512D v2, _M512D v3	float64 vectors
_mm512_mask_maddn132_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, _M512D v3	Multiply, add and negate float64 vectors under mask
_mm512_maddn132_ps	_M512	_M512 v1, _M512 v2, _M512 v3	Multiply, add and negate float32 vectors
_mm512_mask_maddn132_ps	_M512	_M512 v1, _mmask k1, _M512 v2, _M512 v3	Multiply, add and negate float32 vectors under mask
_mm512_maddn213_pd	_M512D	_M512D v1, _M512D v2, _M512D v3	Multiply, add and negate float64 vectors
_mm512_mask_maddn213_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, _M512D v3	Multiply, add and negate float64 vectors under mask
_mm512_maddn213_ps	_M512	_M512 v1, _M512 v2, _M512 v3	Multiply, add and negate float32 vectors
_mm512_mask_maddn213_ps	_M512	_M512 v1, _mmask k1, _M512 v2, _M512 v3	Multiply, add and negate float32 vectors under mask
_mm512_maddn231_pd	_M512D	_M512D v1, _M512D v2, _M512D v3	Multiply, add and negate float64 vectors
_mm512_mask_maddn231_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, _M512D v3	Multiply, add and negate float64 vectors under mask
_mm512_maddn231_ps	_M512	_M512 v1, _M512 v2, _M512 v3	Multiply, add and negate float32 vectors
_mm512_mask_maddn231_ps	_M512	_M512 v1, _mmask k1, _M512 v2, _M512 v3	Multiply, add and negate float32 vectors under mask
_mm512_maxabs_ps	_M512	_M512 v2, _M512 v3	Absolute maximum of float32 vectors
_mm512_mask_maxabs_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, _M512 v3	Absolute maximum of float32 vectors under mask
_mm512_max_pd	_M512D	_M512D v2, _M512D v3	Maximum of float64 vectors
_mm512_mask_max_pd	_M512D	_M512D v1_old, _mmask k1, _M512D v2, _M512D v3	Maximum of float64 vectors under mask
_mm512_max_pi	_M512I	_M512I v2, _M512I v3	Maximum of int32 vectors
_mm512_mask_max_pi	_M512I	_M512I v1_old, _mmask k1, _M512I v2, _M512I v3	Maximum of int32 vectors under mask
_mm512_max_ps	_M512	_M512 v2, _M512 v3	Maximum of float32 vectors
_mm512_mask_max_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, _M512 v3	Maximum of float32 vectors under mask
_mm512_max_pu	_M512I	_M512I v2, _M512I v3	Maximum of uint32 vectors
_mm512_mask_max_pu	_M512I	_M512I v1_old, _mmask k1, _M512I v2, _M512I v3	Maximum of uint32 vectors under mask
_mm512_min_pd	_M512D	_M512D v2, _M512D v3	Minimum of float64 vectors
_mm512_mask_min_pd	_M512D	_M512D v1_old, _mmask k1,	Minimum of float64 vectors under mask

		_M512D v2, M512D v3	
_mm512_min_pi	_M512I	_M512I v2, M512I v3	Minimum of int32 vectors
_mm512_mask_min_pi	_M512I	_M512I v1_old, _mmask k1, _M512I v2, M512I v3	Minimum of int32 vectors under mask
_mm512_min_ps	_M512	M512 v2, M512 v3	Minimum of float32 vectors
_mm512_mask_min_ps	_M512	M512 v1_old, _mmask k1, M512 v2, M512 v3	Minimum of float32 vectors under mask
_mm512_min_pu	_M512I	_M512I v2, _M512I v3	Minimum of uint32 vectors
_mm512_mask_min_pu	_M512I	_M512I v1_old, _mmask k1, _M512I v2, M512I v3	Minimum of uint32 vectors under mask
mm512_mov	M512	M512 v2	Move one vector to another
_mm512_mask_movd	_M512	_M512 v1_old, _mmask k1, M512 v2	Move a doubleword vector to another under mask
_mm512_mask_movq	_M512	M512 v1_old, _mmask k1, M512 v2	Move a quadword vector to another under mask
_mm512_msub132_pd	_M512D	M512D v1, _M512D v2, M512D v3	Multiply and subtract float64 vectors
_mm512_mask_msub132_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, M512D v3	Multiply and subtract float64 vectors under mask
_mm512_msub132_ps	_M512	M512 v1, M512 v2, M512 v3	Multiply and subtract float32 vectors
_mm512_mask_msub132_ps	_M512	M512 v1, _mmask k1, M512 v2, M512 v3	Multiply and subtract float32 vectors under mask
_mm512_msub213_pd	_M512D	_M512D v1, _M512D v2, M512D v3	Multiply and subtract float64 vectors
_mm512_mask_msub213_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, M512D v3	Multiply and subtract float64 vectors under mask
_mm512_msub213_ps	_M512	M512 v1, M512 v2, M512 v3	Multiply and subtract float32 vectors
_mm512_mask_msub213_ps	_M512	M512 v1, _mmask k1, M512 v2, M512 v3	Multiply and subtract float32 vectors under mask
_mm512_msub231_pd	_M512D	M512D v1, _M512D v2, M512D v3	Multiply and subtract float64 vectors
_mm512_mask_msub231_pd	_M512D	_M512D v1, _mmask k1, _M512D v2, M512D v3	Multiply and subtract float64 vectors under mask
_mm512_msub231_ps	_M512	M512 v1, M512 v2, M512 v3	Multiply and subtract float32 vectors
_mm512_mask_msub231_ps	_M512	M512 v1, _mmask k1, M512 v2, M512 v3	Multiply and subtract float32 vectors under mask
_mm512_msubr132_pd	_M512D	M512D v1, _M512D v2, M512D v3	Multiply and subtract float64 vectors
mm512_mask_msubr132_pd	M512D	M512D v1,	Multiply and subtract float64

		<code>_mmask k1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	vectors under mask
<code>_mm512_msubr132_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply and subtract float32 vectors
<code>_mm512_mask_msubr132_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>_mmask k1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply and subtract float32 vectors under mask
<code>_mm512_msubr213_pd</code>	<code>_M512D</code>	<code>M512D v1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	Multiply and subtract float64 vectors
<code>_mm512_mask_msubr213_pd</code>	<code>_M512D</code>	<code>_M512D v1,</code> <code>_mmask k1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	Multiply and subtract float64 vectors under mask
<code>_mm512_msubr213_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply and subtract float32 vectors
<code>_mm512_mask_msubr213_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>_mmask k1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply and subtract float32 vectors under mask
<code>_mm512_msubr231_pd</code>	<code>_M512D</code>	<code>_M512D v1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	Multiply and subtract float64 vectors
<code>_mm512_mask_msubr231_pd</code>	<code>_M512D</code>	<code>_M512D v1,</code> <code>_mmask k1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	Multiply and subtract float64 vectors under mask
<code>_mm512_msubr231_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply and subtract float32 vectors
<code>_mm512_mask_msubr231_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>_mmask k1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply and subtract float32 vectors under mask
<code>_mm512_msubr23c1_pd</code>	<code>_M512D</code>	<code>_M512D v2,</code> <code>M512D v3</code>	Multiply float64 vectors and subtract from 1
<code>_mm512_mask_msubr23c1_pd</code>	<code>_M512D</code>	<code>_M512D v1_old,</code> <code>_mmask k1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	Multiply float64 vectors and subtract from 1 under mask
<code>_mm512_msubr23c1_ps</code>	<code>_M512</code>	<code>_M512 v2,</code> <code>M512 v3</code>	Multiply float32 vectors and subtract from 1
<code>_mm512_mask_msubr23c1_ps</code>	<code>_M512</code>	<code>_M512 v1_old,</code> <code>_mmask k1,</code> <code>_M512 v2,</code> <code>M512 v3</code>	Multiply float32 vectors and subtract from 1 under mask
<code>_mm512_mulh_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	Multiply int32 vectors and store high result
<code>_mm512_mask_mulh_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Multiply int32 vectors and store high result under mask
<code>_mm512_mulh_pu</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	Multiply uint32 vectors and store high result
<code>_mm512_mask_mulh_pu</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Multiply uint32 vectors and store high result under mask
<code>_mm512_mull_pi</code>	<code>_M512I</code>	<code>_M512I v2,</code> <code>M512I v3</code>	Multiply int32 vectors and store low result
<code>_mm512_mask_mull_pi</code>	<code>_M512I</code>	<code>_M512I v1_old,</code> <code>_mmask k1,</code> <code>_M512I v2,</code> <code>M512I v3</code>	Multiply int32 vectors and store low result under mask
<code>_mm512_mul_pd</code>	<code>_M512D</code>	<code>_M512D v2,</code> <code>M512D v3</code>	Multiply float64 vectors
<code>_mm512_mask_mul_pd</code>	<code>_M512D</code>	<code>_M512D v1_old,</code> <code>_mmask k1,</code> <code>_M512D v2,</code> <code>M512D v3</code>	Multiply float64 vectors under mask
<code>_mm512_mul_ps</code>	<code>_M512</code>	<code>_M512 v2,</code> <code>M512 v3</code>	Multiply float32 vectors

_mm512_mask_mul_ps	_M512	_M512 v1_old, __mmask k1, _M512 v2, M512 v3	Multiply float32 vectors under mask
_mm512_sbb_pi	_M512I	_M512I v1, __mmask k2, _M512I v3, __mmask *k2 res	Subtract int32 vectors with borrow
_mm512_mask_sbb_pi	_M512I	_M512I v1, __mmask k1, __mmask k2, _M512I v3, __mmask *k2 res	Subtract int32 vectors with borrow under mask
_mm512_sbb_r_pi	_M512I	_M512I v1, __mmask k2, _M512I v3, __mmask *k2 res	Reverse subtract int32 vectors with borrow
_mm512_mask_sbb_r_pi	_M512I	_M512I v1, __mmask k1, __mmask k2, _M512I v3, __mmask *k2 res	Reverse subtract int32 vectors with borrow under mask
_mm512_sub_pd	_M512D	_M512D v2, M512D v3	Subtract float64 vectors
_mm512_mask_sub_pd	_M512D	_M512D v1_old, __mmask k1, _M512D v2, M512D v3	Subtract float64 vectors under mask
_mm512_sub_pi	_M512I	_M512I v2, M512I v3	Subtract int32 vectors
_mm512_mask_sub_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, M512I v3	Subtract int32 vectors under mask
_mm512_sub_ps	_M512	_M512 v2, M512 v3	Subtract float32 vectors
_mm512_mask_sub_ps	_M512	_M512 v1_old, __mmask k1, _M512 v2, M512 v3	Subtract float32 vectors under mask
_mm512_sub_r_pd	_M512D	_M512D v2, M512D v3	Reverse subtract float64 vectors
_mm512_mask_sub_r_pd	_M512D	_M512D v1_old, __mmask k1, _M512D v2, M512D v3	Reverse subtract float64 vectors under mask
_mm512_sub_r_pi	_M512I	_M512I v2, M512I v3	Reverse subtract int32 vectors
_mm512_mask_sub_r_pi	_M512I	_M512I v1_old, __mmask k1, _M512I v2, M512I v3	Reverse subtract int32 vectors under mask
_mm512_sub_r_ps	_M512	_M512 v2, M512 v3	Reverse subtract float32 vectors
_mm512_mask_sub_r_ps	_M512	_M512 v1_old, __mmask k1, _M512 v2, M512 v3	Reverse subtract float32 vectors under mask
_mm512_sub_rsetb_pi	_M512I	_M512I v1, _M512I v3, __mmask *borrow	Subtract int32 vectors and set borrow
_mm512_mask_sub_rsetb_pi	_M512I	_M512I v1, __mmask k1, __mmask k2_old, _M512I v3, __mmask *borrow	Subtract int32 vectors and set borrow under mask
_mm512_subsetb_pi	_M512I	_M512I v1, _M512I v3, __mmask *borrow	Subtract int32 vectors and set borrow
_mm512_mask_subsetb_pi	_M512I	_M512I v1, __mmask k1, __mmask k2_old, _M512I v3, __mmask *borrow	Subtract int32 vectors and set borrow under mask
Compress/Expand			

		float e6, float e5, float e4, float e3, float e2, float e1, float e0	
_mm512_set_16to16_pi	_M512I	int e15, int e14, int e13, int e12, int e11, int e10, int e9, int e8, int e7, int e6, int e5, int e4, int e3, int e2, int e1, int e0	Return int32 vector e15 e14 e13 ... e1 e0 (v15=e15, v14=e14, ..., v0=e0)
_mm512_set_8to8_pd	_M512D	double e7, double e6, double e5, double e4, double e3, double e2, double e1, double e0	Return float64 vector e7 e6 e5 ... e1 e0 (v7=e7, v6=e6, ..., v0=e0)
_mm512_set_8to8_pq	_M512I	int64_t e7, int64_t e6, int64_t e5, int64_t e4, int64_t e3, int64_t e2, int64_t e1, int64_t e0	Return int64 vector e7 e6 e5 ... e1 e0 (v7=e7, v6=e6, ..., v0=e0)
Math			
_mm512_round_ps	_M512	_M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_EXP_ADJ_ENUM expadj	Round float32 vector
_mm512_mask_round_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, _MM_ROUND_MODE_ENUM rc, _MM_EXP_ADJ_ENUM expadj	Round float32 vector under mask
_mm512_scale_ps	_M512	_M512 v2, _M512 v3	Scale float32 vectors
_mm512_mask_scale_ps	_M512	_M512 v1_old, _mmask k1, _M512 v2, _M512 v3	Scale float32 vectors under mask
mm512_acos_pd	M512D	M512D v1	Arc cosine of a float64 vector
_mm512_mask_acos_pd	_M512D	_M512D v0_old, _mmask k1, _M512D v1	Arc cosine of a float64 vector under mask
mm512_acos_ps	M512	M512 v1	Arc cosine of a float32 vector
_mm512_mask_acos_ps	_M512	_M512 v0_old, _mmask k1, _M512 v1	Arc cosine of a float32 vector under mask
mm512_asin_pd	M512D	M512D v1	Arc sine of a float64 vector
_mm512_mask_asin_pd	_M512D	_M512D v0_old, _mmask k1, _M512D v1	Arc sine of a float64 vector under mask
mm512_asin_ps	M512	M512 v1	Arc sine of a float32 vector
_mm512_mask_asin_ps	_M512	_M512 v0_old, _mmask k1, _M512 v1	Arc sine of a float32 vector under mask
_mm512_atan2_pd	_M512D	_M512D v1, _M512D v2	Arc tangent of float64 vectors
_mm512_mask_atan2_pd	_M512D	_M512D v0_old, _mmask k1, _M512D v1, _M512D v2	Arc tangent of float64 vectors under mask
_mm512_atan2_ps	_M512	_M512 v1, _M512 v2	Arc tangent of float32 vectors
_mm512_mask_atan2_ps	_M512	_M512 v0_old, _mmask k1, _M512 v1, _M512 v2	Arc tangent of float32 vectors under mask
mm512_atan_pd	M512D	M512D v1	Arc tangent of a float64 vector
_mm512_mask_atan_pd	_M512D	_M512D v0_old, _mmask k1, _M512D v1	Arc tangent of a float64 vector under mask
mm512_atan_ps	M512	M512 v1	Arc tangent of a float32 vector
_mm512_mask_atan_ps	_M512	_M512 v0_old, _mmask k1, _M512 v1	Arc tangent of a float32 vector under mask
_mm512_ceil_pd	_M512D	_M512D v1	Round float64 vector to nearest upper integer
_mm512_mask_ceil_pd	_M512D	_M512D v0_old, _mmask k1, _M512D v1	Round float64 vector to nearest upper integer under mask
_mm512_ceil_ps	_M512	_M512 v1	Round float32 vector to nearest upper integer
mm512_mask_ceil_ps	M512	M512 v0 old,	Round float32 vector to nearest

		<code>_mmask k1,</code> <code>M512 v1</code>	upper integer under mask
<code>mm512_cos_pd</code>	<code>M512D</code>	<code>M512D v1</code>	Cosine of a float64 vector
<code>_mm512_mask_cos_pd</code>	<code>_M512D</code>	<code>_M512D v0_old,</code> <code>_mmask k1,</code> <code>M512D v1</code>	Cosine of a float64 vector under mask
<code>mm512_cos_ps</code>	<code>M512</code>	<code>M512 v1</code>	Cosine of a float32 vector
<code>_mm512_mask_cos_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1</code>	Cosine of a float32 vector under mask
<code>_mm512_cosh_pd</code>	<code>_M512D</code>	<code>_M512D v1</code>	Hyperbolic cosine of a float64 vector
<code>_mm512_mask_cosh_pd</code>	<code>_M512D</code>	<code>_M512D v0_old,</code> <code>_mmask k1,</code> <code>M512D v1</code>	Hyperbolic cosine of a float64 vector under mask
<code>_mm512_cosh_ps</code>	<code>_M512</code>	<code>_M512 v1</code>	Hyperbolic cosine of a float32 vector
<code>_mm512_mask_cosh_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1</code>	Hyperbolic cosine of a float32 vector under mask.
<code>_mm512_exp2_pd</code>	<code>_M512D</code>	<code>_M512D v1</code>	Exponential base-2 of a float64 vector
<code>_mm512_mask_exp2_pd</code>	<code>_M512D</code>	<code>_M512D v0_old,</code> <code>_mmask k1,</code> <code>M512D v1</code>	Exponential base-2 of a float64 vector under mask
<code>_mm512_exp2_ps</code>	<code>_M512</code>	<code>_M512 v1</code>	Exponential base-2 of a float32 vector
<code>_mm512_mask_exp2_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1</code>	Exponential base-2 of a float32 vector under mask
<code>_mm512_exp_pd</code>	<code>_M512D</code>	<code>_M512D v1</code>	Exponential base-e of a float64 vector
<code>_mm512_mask_exp_pd</code>	<code>_M512D</code>	<code>_M512D v0_old,</code> <code>_mmask k1,</code> <code>M512D v1</code>	Exponential base-e of a float64 vector under mask.
<code>_mm512_exp_ps</code>	<code>_M512</code>	<code>_M512 v1</code>	Exponential base-e of a float32 vector
<code>_mm512_mask_exp_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1</code>	Exponential base-e of a float32 vector under mask
<code>_mm512_floor_pd</code>	<code>_M512D</code>	<code>_M512D v1</code>	Round float64 vector to nearest lower integer
<code>_mm512_mask_floor_pd</code>	<code>_M512D</code>	<code>_M512D v0_old,</code> <code>_mmask k1,</code> <code>M512D v1</code>	Round float64 vector to nearest lower integer under mask
<code>_mm512_floor_ps</code>	<code>_M512</code>	<code>_M512 v1</code>	Round float32 vector to nearest lower integer
<code>_mm512_mask_floor_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1</code>	Round float32 vector to nearest lower integer under mask
<code>_mm512_hypot_pd</code>	<code>_M512D</code>	<code>_M512D v1,</code> <code>M512D v2</code>	Hypotenuse of float64 vectors
<code>_mm512_mask_hypot_pd</code>	<code>_M512D</code>	<code>_M512D v0_old,</code> <code>_mmask k1,</code> <code>M512D v1,</code> <code>M512D v2</code>	Hypotenuse of float64 vectors under mask
<code>_mm512_hypot_ps</code>	<code>_M512</code>	<code>_M512 v1,</code> <code>M512 v2</code>	Hypotenuse of float32 vectors
<code>_mm512_mask_hypot_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1,</code> <code>M512 v2</code>	Hypotenuse of float32 vectors under mask
<code>_mm512_div_pi</code>	<code>_M512I</code>	<code>M512I v1,</code> <code>M512I v2</code>	Quotient of int32 vectors
<code>_mm512_mask_div_pi</code>	<code>_M512I</code>	<code>M512I v0_old,</code> <code>_mmask k1,</code> <code>M512I v1,</code> <code>M512I v2</code>	Quotient of int32 vectors under mask
<code>_mm512_div_ps</code>	<code>_M512</code>	<code>M512 v1,</code> <code>M512 v2</code>	Quotient of float32 vectors
<code>_mm512_mask_div_ps</code>	<code>_M512</code>	<code>_M512 v0_old,</code> <code>_mmask k1,</code> <code>M512 v1,</code> <code>M512 v2</code>	Quotient of float32 vectors under mask
<code>mm512_div_pd</code>	<code>M512D</code>	<code>M512D v1,</code>	Quotient of float64 vectors

		M512D v2	
_mm512_mask_div_pd	_M512D	_M512D v0_old, _mmask k1, _M512D v1, M512D v2	Quotient of float64 vectors under mask
_mm512_div_pu	_M512I	_M512I v1, M512I v2	Quotient of uint32 vectors
_mm512_mask_div_pu	_M512I	_M512I v0_old, _mmask k1, _M512I v1, M512I v2	Quotient of uint32 vectors under mask
_mm512_rsqrt_pd	_M512D	_M512D v1	Reciprocal square root of float64 vector
_mm512_mask_rsqrt_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Reciprocal square root of float64 vector under mask
_mm512_rsqrt_ps	_M512	_M512 v1	Reciprocal square root of float32 vector
_mm512_mask_rsqrt_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Reciprocal square root of float32 vector under mask
mm512_recip_pd	M512D	M512D v1	Reciprocal of float64 vector
_mm512_mask_recip_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Reciprocal of float64 vector under mask
mm512_recip_ps	M512	M512 v1	Reciprocal of float32 vector
_mm512_mask_recip_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Reciprocal of float32 vector under mask
_mm512_rem_pi	_M512I	_M512I v1, _M512I v2	Remainder of the division of int32 vectors
_mm512_mask_rem_pi	_M512I	_M512I v0_old, _mmask k1, _M512I v1, M512I v2	Remainder of the division of int32 vectors under mask
_mm512_rem_pu	_M512I	_M512I v1, M512I v2	Remainder of the division of two uint32 vectors
_mm512_mask_rem_pu	_M512I	_M512I v0_old, _mmask k1, _M512I v1, M512I v2	Remainder of the division of two uint32 vectors under mask
_mm512_log10_pd	_M512D	_M512D v1	Logarithm base-10 of float64 vector
_mm512_mask_log10_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Logarithm base-10 of float64 vector under mask
_mm512_log10_ps	_M512	_M512 v1	Logarithm base-10 of float32 vector
_mm512_mask_log10_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Logarithm base-10 of float32 vector under mask
_mm512_log2_pd	_M512D	_M512D v1	Logarithm base-2 of float64 vector
_mm512_mask_log2_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Logarithm base-2 of float64 vector under mask
_mm512_log2_ps	_M512	_M512 v1	Logarithm base-2 of float32 vector
_mm512_mask_log2_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Logarithm base-2 of float32 vector under mask
_mm512_log_pd	_M512D	_M512D v1	Logarithm base-e of float64 vector
_mm512_mask_log_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Logarithm base-e of float64 vector under mask
_mm512_log_ps	_M512	_M512 v1	Logarithm base-e of float32 vector
_mm512_mask_log_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Logarithm base-e of float32 vector under mask
_mm512_pow_pd	_M512D	_M512D v1, M512D v2	Float64 vector raised to the power of another float64 vector
_mm512_mask_pow_pd	_M512D	_M512D v0_old, _mmask k1,	Float64 vector raised to the power of another float64 vector

		_M512D v1, M512D v2	under mask
_mm512_pow_ps	_M512	_M512 v1, M512 v2	Float32 vector raised to the power of another float32 vector
_mm512_mask_pow_ps	_M512	_M512 v0_old, _mmask k1, _M512 v1, M512 v2	Float32 vector raised to the power of another float32 vector under mask
mm512_sin_pd	M512D	M512D v1	Sine of a float64 vector
_mm512_mask_sin_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Sine of a float64 vector under mask
mm512_sin_ps	M512	M512 v1	Sine of a float32 vector
_mm512_mask_sin_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Sine of a float32 vector under mask
_mm512_sinh_pd	_M512D	_M512D v1	Hyperbolic sine of a float64 vector.
_mm512_mask_sinh_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Hyperbolic sine of a float64 vector under mask
_mm512_sinh_ps	_M512	_M512 v1	Hyperbolic sine of a float32 vector
_mm512_mask_sinh_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Hyperbolic sine of a float32 vector under mask
mm512_sqrt_pd	M512D	M512D v1	Square root of float64 vector
_mm512_mask_sqrt_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1a	Square root of float64 vector under mask.
mm512_sqrt_ps	M512	M512 v1	Square root of float32 vector
_mm512_mask_sqrt_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Square root of float32 vector under mask
mm512_tan_pd	M512D	M512D v1	Tangent of float64 vector
_mm512_mask_tan_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Tangent of float64 vector under mask
mm512_tan_ps	M512	M512 v1	Tangent of float32 vector
_mm512_mask_tan_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Tangent of float32 vector under mask
_mm512_tanh_pd	_M512D	_M512D v1	Hyperbolic tangent of float64 vector
_mm512_mask_tanh_pd	_M512D	_M512D v0_old, _mmask k1, M512D v1	Hyperbolic tangent of float64 vector under mask
_mm512_tanh_ps	_M512	_M512 v1	Hyperbolic tangent of float32 vector
_mm512_mask_tanh_ps	_M512	_M512 v0_old, _mmask k1, M512 v1	Hyperbolic tangent of float32 vector under mask
_mm512_reduce_add_ps	float	_M512 v2	Sum of all elements of float32 vector
_mm512_mask_reduce_add_ps	float	_mmask k1, _M512 v2	Sum of all elements of float32 vector under mask. Returns 0 if mask =0.
_mm512_reduce_add_pd	double	_M512D v2	Sum of all elements of float64 vector
_mm512_mask_reduce_add_pd	double	_mmask k1, _M512D v2	Sum of all elements of float64 vector under mask. Returns 0 if mask = 0.
_mm512_reduce_add_pi	int	_M512I v2	Sum of all elements of int32 vector
_mm512_mask_reduce_add_pi	int	_mmask k1, _M512I v2	Sum of all elements of int32 vector under mask. Returns 0 if mask = 0.
_mm512_reduce_mul_ps	float	_M512 v2	Product of all elements of float32 vector
_mm512_mask_reduce_mul_ps	float	_mmask k1, _M512 v2	Product of all elements of float32 vector under mask. Returns 1 if mask = 0.
_mm512_reduce_mul_pd	double	_M512D v2	Product of all elements of float64 vector
mm512_mask_reduce_mul_pd	double	_mmask k1,	Product of all elements of

		_M512D v2	float64 vector under mask. Returns 1 if mask = 0.
_mm512_reduce_mul_pi	int	_M512I v2	Product of all elements of int32 vector
_mm512_mask_reduce_mul_pi	int	_mmask k1, _M512I v2	Product of all elements of int32 vector under mask. Returns 1 if mask = 0.
_mm512_reduce_min_ps	float	_M512 v2	Minimum of all elements of float32 vector
_mm512_mask_reduce_min_ps	float	_mmask k1, _M512 v2	Minimum of all elements of float32 vector under mask. Returns FLT_MAX if mask = 0
_mm512_reduce_min_pd	double	_M512D v2	Minimum of all elements of float64 vector
_mm512_mask_reduce_min_pd	double	_mmask k1, _M512D v2	Minimum of all elements of float64 vector under mask. Returns DBL_MAX if mask = 0
_mm512_reduce_min_pi	int	_M512I v2	Minimum of all elements of int32 vector
_mm512_mask_reduce_min_pi	int	_mmask k1, _M512I v2	Minimum of all elements of int32 vector under mask. Returns 0xffffffff if mask = 0
_mm512_reduce_max_ps	float	_M512 v2	Maximum of all elements of float32 vector
_mm512_mask_reduce_max_ps	float	_mmask k1, _M512 v2	Maximum of all elements of float32 vector under mask. Returns FLT_MIN if mask = 0
_mm512_reduce_max_pd	double	_M512D v2	Maximum of all elements of float64 vector
_mm512_mask_reduce_max_pd	double	_mmask k1, _M512D v2	Maximum of all elements of float64 vector under mask. Returns DBL_MIN if mask = 0
_mm512_reduce_max_pi	int	_M512I v2	Maximum of all elements of int32 vector
_mm512_mask_reduce_max_pi	int	_mmask k1, _M512I v2	Maximum of all elements of int32 vector under mask. Returns 0x80000000 if mask = 0
_mm512_reduce_or_pi	int	_M512I v2	Logical OR of all elements of int32 vector
_mm512_mask_reduce_or_pi	int	_mmask k1, _M512I v2	Logical OR of all elements of int32 vector under mask. Returns 0 if mask = 0
_mm512_reduce_and_pi	int	_M512I v2	Logical AND of all elements of int32 vector
_mm512_mask_reduce_and_pi	int	_mmask k1, _M512I v2	Logical AND of all elements of int32 vector under mask. Returns 0xffffffff if mask = 0
Scalar			
_mm_bitinterleavell_16	unsigned int	unsigned short r1, unsigned short r2	1:1 bit interleave
_mm_bitinterleavell_32	unsigned int	unsigned int r1, unsigned int r2	1:1 bit interleave
_mm_bitinterleavell_64	uint64_t	uint64_t r1, uint64_t r2	1:1 bit interleave
_mm_bitinterleave21_16	unsigned short	unsigned short r1, unsigned short r2	2:1 bit interleave
_mm_bitinterleave21_32	unsigned int	unsigned int r1, unsigned int r2	2:1 bit interleave
_mm_bitinterleave21_64	uint64_t	uint64_t r1, uint64_t r2	2:1 bit interleave
mm bsff 16	short	unsigned short r1	Fast bit scan forward
mm bsff 32	int	unsigned int r1	Fast bit scan forward
mm bsff 64	int64_t	uint64_t r1	Fast bit scan forward
_mm_bsfi_16	short	short r1, unsigned short r2	Bit scan forward initialized
_mm_bsfi_32	int	int r1, unsigned int r2	Bit scan forward initialized
_mm_bsfi_64	int64_t	int64_t r1, uint64_t r2	Bit scan forward initialized
mm bsrff 16	short	unsigned short r1	Fast bit scan reverse
mm bsrff 32	int	unsigned int r1	Fast bit scan reverse
mm bsrff 64	int64_t	uint64_t r1	Fast bit scan reverse
_mm_bsri_16	short	short r1, unsigned short r2	Bit scan reverse initialized

_mm_bsri_32	int	int r1, unsigned int r2	Bit scan reverse initialized
_mm_bsri_64	int64_t	int64_t r1, uint64_t r2	Bit scan reverse initialized
_mm_countbits_16	unsigned short	unsigned short r1	Bit population count
_mm_countbits_32	unsigned int	unsigned int r1	Bit population count
mm_countbits_64	uint64_t	uint64_t r1	Bit population count
_mm_insertfield_16	unsigned short	unsigned short r1, unsigned short r2, _MM_BITPOSITION16_ENUM rotation, _MM_BITPOSITION16_ENUM bit_idx_low, _MM_BITPOSITION16_ENUM bit_idx_high	Rotate and bitfield-insert
_mm_insertfield_32	unsigned int	unsigned int r1, unsigned int r2, _MM_BITPOSITION32_ENUM rotation, _MM_BITPOSITION32_ENUM bit_idx_low, _MM_BITPOSITION32_ENUM bit_idx_high	Rotate and bitfield-insert
_mm_insertfield_64	uint64_t	uint64_t r1, uint64_t r2, _MM_BITPOSITION64_ENUM rotation, _MM_BITPOSITION64_ENUM bit_idx_low, _MM_BITPOSITION64_ENUM bit_idx_high	Rotate and bitfield-insert
_mm_quadmask16_16	unsigned short	unsigned short r1	Set per-quad mask
_mm_quadmask16_32	unsigned int	unsigned int r1	Set per-quad mask
mm_quadmask16_64	uint64_t	uint64_t r1	Set per-quad mask
_mm_rotatefield_16	unsigned short	unsigned short r2, _MM_BITPOSITION16_ENUM rotation, _MM_BITPOSITION16_ENUM bit_idx_low, _MM_BITPOSITION16_ENUM bit_idx_high	Rotate and mask
_mm_rotatefield_32	unsigned int	unsigned int r2, _MM_BITPOSITION32_ENUM rotation, _MM_BITPOSITION32_ENUM bit_idx_low, _MM_BITPOSITION32_ENUM bit_idx_high	Rotate and mask
_mm_rotatefield_64	uint64_t	uint64_t r2, _MM_BITPOSITION64_ENUM rotation, _MM_BITPOSITION64_ENUM bit_idx_low, _MM_BITPOSITION64_ENUM bit_idx_high	Rotate and mask.
Prefetch			
_mm_vprefetch1	void	void *, _MM_PREFETCH_HINT_ENUM	Prefetch an L1 cache line
_mm_vprefetch2	void	void *, _MM_PREFETCH_HINT_ENUM	Prefetch an L2 cache line
Vector Mask			
mm512_vkand	mmask	mmask k1, mmask k2	
mm512_vkandn	mmask	mmask k1, mmask k2	
mm512_vkandnr	mmask	mmask k1, mmask k2	
mm512_vkmov	mmask	mmask k1	
mm512_vkmovlhb	mmask	mmask k1, mmask k2	
mm512_vknot	mmask	mmask k1	
mm512_vkor	mmask	mmask k1, mmask k2	
mm512_vkortestz	bool	mmask k1, mmask k2	
mm512_vkortestc	bool	mmask k1, mmask k2	
mm512_vkxnor	mmask	mmask k1, mmask k2	
mm512_vkxor	mmask	mmask k1, mmask k2	
mm512_vkswapb	mmask	mmask k1, mmask k2	
mm512_mask2int	int	mmask k1	
mm512_int2mask	mmask	int k1	